

# HCC OAL for SYSBIOS User Guide

Version 1.10

For use with OAL for SYS/BIOS™ versions 1.02 and above

**Date:** 24-Feb-2016 12:55

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
- Change History \_\_\_\_\_ 5
- Source File List \_\_\_\_\_ 6
- Configuration File \_\_\_\_\_ 6
- Source Files \_\_\_\_\_ 6
- Version File \_\_\_\_\_ 6
- Configuration Options \_\_\_\_\_ 7
- Implementation Notes \_\_\_\_\_ 8

# 1 System Overview

## 1.1 Introduction

---

This guide is for those who want to use HCC Embedded's OS Abstraction Layer (OAL) for their developments in embedded systems which use the SYS/BIOS™ operating system from Texas Instruments Incorporated.

The HCC OAL is an abstraction of a Real Time Operating System (RTOS). It defines how HCC software requires an RTOS to behave and its Application Programming Interface (API) defines the functions it requires. Most HCC systems and modules use one or more components of the OAL.

HCC has ported its OAL to SYS/BIOS™, in the process creating "hooks" which call SYS/BIOS™ functions from the HCC abstractions. Once you unzip the files from the **oal\_os\_sysbios** package into the **oal/os** folder in the source tree, these files automatically call the correct functions.

The OAL API defines functions for handling the following elements:

- Tasks.
- Events – these are used as a signaling mechanism, both between tasks, and from asynchronous sources such as Interrupt Service Routines (ISRs) to tasks.
- Mutexes – these guarantee that, while one task is using a particular resource, no other task can preempt it and use the same resource.
- Interrupt Service Routines (ISRs) – in SYS/BIOS™ ISRs are platform-specific.

---

## 1.2 Feature Check

---

The main features of the module are the following:

- It conforms to the HCC Advanced Embedded Framework.
- It is integrated with the HCC OS Abstraction Layer (OAL).
- It allows all HCC middleware to run with the SYS/BIOS™ RTOS.

## 1.3 Packages and Documents

---

### Packages

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

Package	Description
oal_base	The OAL base package.
oal_os_sysbios	The OAL for SYS/BIOS™ package. Unzip the files from this package into the <b>oal/os</b> folder in the source tree.

### Documents

For an overview of HCC RTOS software, refer to the [Product Information](#) section of 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 describes the base OAL package, defining the standard functions that must be provided by an RTOS. Use this as your reference to global configuration options and the API.

#### HCC OAL for SYS/BIOS User Guide

This is this document.

## 1.4 Change History

---

This section includes recent changes to this product. For a list of all the changes, refer to the file **src/history/oal/oal\_os\_sysbios.txt** in the distribution package.

Version	Changes
1.02	Added <i>t_oal_ret</i> return code Added API function <b>oal_event_clear()</b> .
1.01	Initial release.

## 2 Source File List

This section lists and describes all the source code files included in the system. These files follow HCC Embedded's 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 files except the configuration file.

### 2.1 Configuration File

The file `src/config/config_oal_os.h` contains [configuration options](#) specific to the system. Configure these as required. (Global configuration parameters are controlled by the base package's configuration file.)

### 2.2 Source Files

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

File	Description
<code>oalp_defs.h</code>	System defines header file.
<code>oalp_event.c</code>	Event functions source code.
<code>oalp_event.h</code>	Event functions header file.
<code>oalp_isr.c</code>	ISR functions source code.
<code>oalp_isr.h</code>	ISR functions header file.
<code>oalp_mutex.c</code>	Mutex functions source code.
<code>oalp_mutex.h</code>	Mutex functions header file.
<code>oalp_task.c</code>	Task functions source code.
<code>oalp_task.h</code>	Task functions header file.

### 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.

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

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

By default these are respectively 14, 10, 7, 4, and 2.

**OAL\_EVENT\_FLAG**

The event flag to use for user tasks invoking internal functions. The default is 0x100.

**OAL\_TICK\_RATE**

The tick rate in ms. Set this to the SYS/BIOS™ tick rate. The default is 1.

**OAL\_ISR\_COUNT**

The maximum number of interrupt sources. The default is 6.

## 4 Implementation Notes

The RTOS elements are implemented as follows.

### **Events**

There are no rules governing events.

### **Mutexes**

There are no rules governing mutexes.

### **Tasks**

There are no rules governing tasks.

### **ISRs**

The configuration option `OAL_ISR_COUNT` sets the maximum number of interrupts supported by the OAL.

### **Ticks**

Set the configuration option `OAL_TICK_RATE` to the SYS/BIOS™ tick rate.