



# HCC OAL for SYSBIOS User Guide

Version 1.30

For use with OAL for SYS/BIOS™ versions 1.03 and above

Exported on 06/08/2018

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	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 Files .....	8
2.3	Version File .....	8
3	Configuration Options .....	9
4	Implementation Notes.....	10

# 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 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:

- Conforms to the HCC Advanced Embedded Framework.
- Integrated with the HCC OS Abstraction Layer (OAL).
- 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
<b>oal_base</b>	The OAL base package.
<b>oal_os_sysbios</b>	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, 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 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 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\\_sysbios](#).

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	2018-06-08	1.03	Corrected text on OAL_EVENT_FLAG configuration option.
1.20	2017-06-28	1.03	New <i>Change History</i> format.
1.10	2016-02-24	1.02	Modified <i>Feature Check</i> section.
1.00	2014-10-21	1.01	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 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

Set the SYS/BIOS™ 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\_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**

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.

### **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.