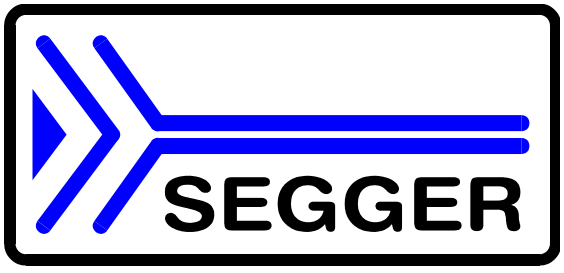
embOS
C-SPY Plug-in

embOS plug-in for the
IAR C-SPY® Debugger

Version 2.00e
Manual Rev. 1



A product of SEGGER Microcontroller Systeme GmbH

Disclaimer

Specifications written in this document are believed to be accurate, but are not guaranteed to be entirely free of error. The information in this manual is subject to change for functional or performance improvements without notice. Please make sure your manual is the latest edition. While the information herein is assumed to be accurate, SEGGER MICROCONTROLLER SYSTEME GmbH (the manufacturer) assumes no responsibility for any errors or omissions. The manufacturer makes and you receive no warranties or conditions, express, implied, statutory or in any communication with you. The manufacturer specifically disclaims any implied warranty of merchantability or fitness for a particular purpose.

Copyright notice

Trademarks

Names mentioned in this manual may be trademarks of their respective companies. Brand and product names are trademarks or registered trademarks of their respective holders.

Contact address

SEGGER Microcontroller Systeme GmbH
Heinrich-Hertz-Str. 5
D-40721 Hilden
Germany
Tel.+49 2103-2878-0
Fax.+49 2103-2878-28
Email: support@segger.com
Internet: http://www.segger.com

Manual versions

Manual version	Date	By	Explanation
1.0	070707	TW	Initial version for plugin V2.

Software versions

Software version	Date	By	Explanation
2.00e	070707	AW	Initial version for IAR Embedded Workbench V5

Table of Contents

Chapter 1

Introduction

This chapter gives a short overview about the embOS C-Spy plug-in for IAR Embedded Workbench®.

1.1 Overview

1.1.1 embOS C-Spy Plug-in for IAR Embedded Workbench

SEGGER's embOS C-Spy plug-in for IAR Embedded Workbench provides embOS-awareness during debugging sessions. This enables you to inspect the state of several embOS primitives such as the task list, resource semaphores, mailboxes, and timers.

1.1.2 embOS

embOS is a real-time operating system for embedded applications designed to offer the benefits of a fully-fledged multitasking system at minimum cost. The kernel is fully interruptible and so efficient that embOS can be used in very time critical situations. The memory footprint in both RAM and ROM is so small that embOS can be used in single-chip applications, leaving maximum room for the user-program.

1.1.3 IAR Embedded Workbench

IAR Embedded Workbench is a set of development tools for building and debugging embedded applications using assembler, C and C++. It provides a completely integrated development environment that includes a project manager, editor, build tools and the C-SPY debugger. IAR Embedded Workbench supports a wide range of microcontrollers and cores from different chip manufacturers. It offers the same intuitive user interface regardless of which microcontroller you have chosen to work with—coupled with general and target-specific support for each chip.

1.2 Requirements

To use the embOS C-Spy plug-in you need a version of IAR Embedded Workbench installed and a debug target which uses embOS. Specifically:

- An embOS version 3.28 or higher is required for complete compatibility. Older embOS versions use different internal structures and the C-Spy plug-in is therefore of limited use with version prior to 3.28.
- An IAR Embedded Workbench IDE with a C-SPY debugger version 5.0 or higher is required for the plug-in V2.

1.3 Supported CPUs

The embOS C-Spy plug-in works with 16-bit or 32-bit CPUs in little- or big-endian mode supported by embOS, but due to limited testing, support can only be granted for the CPUs listed below.

- Any ARM7 / ARM9 CPU
- Atmel AVR / ATMega
- National CR16C
- Renesas H8/H8S
- Renesas M16C
- Renesas M16C80
- Renesas M32C
- Renesas R32C
- Renesas R8C
- NEC V850
- NEC 78/K0
- TI MSP430

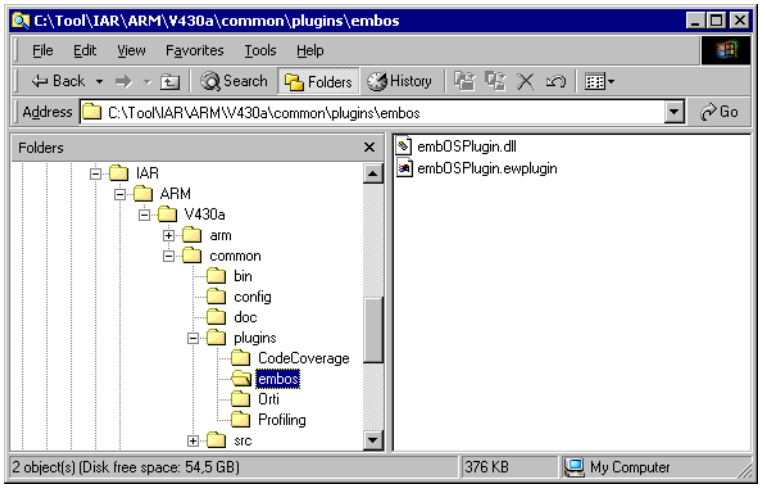
Chapter 2

Installation

This chapter describes the installation steps required to use the embOS C-Spy plug-in.

2.1 Installation Procedure

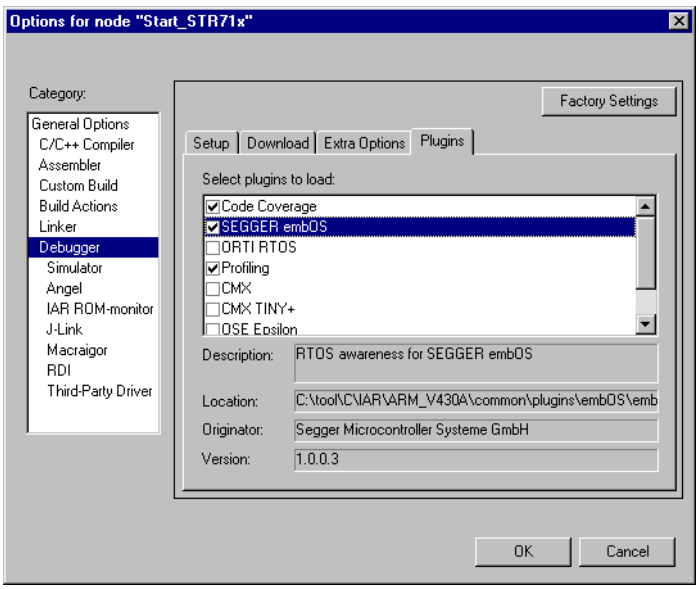
Normally there is no installation required, because the IAR Embedded Workbench comes witrh the plug-in already installed. If for some reason you want to uptade the plug-in, you have to replace two files.
The installation procedure is very straightforward because it only requires you to copy the contents of the embOS C-Spy plug-in package into the IAR CPU specific plug-in folder for `rtos` plug-ins. The directory structure may look like this:



If not already delivered with the IAR Embedded Workbench IDE, create a directory `embOS` below the CPU specific `plugin\rtos\` folder and copy the files from the embOS folder which comes with the plugin into that folder in your IAR installation directory. Then restart the IAR Embedded Workbench IDE.

2.2 Configuration

By default, the embOS C-Spy plug-in is not loaded during debugging. For each project configuration you have to explicitly enable the plug-in in the debugger section of the project options:



The embOS C-Spy plug-in is now available in debugging sessions and may be accessed from the main menu.

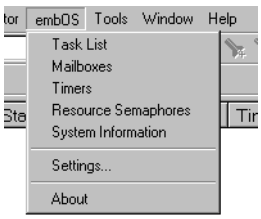
Chapter 3

Getting started

This chapter describes the embOS C-Spy plug-in and its capabilities in greater detail.

3.1 Overview

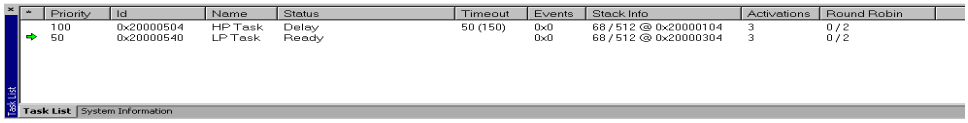
During your debugging session, the embOS C-Spy plug-in is accessible from the IAR Embedded Workbench IDE **main** menu. Note that if you are not running a debugging session, there is no **embOS menu** item available.



From the menu you may activate the individual windows that provide embOS related information. The sections below describe these individual windows. The amount of information available depends on the embOS build used during debugging. If a certain part is not available, the respective menu item is either greyed out or the window column shows a **N/A**.

3.2 Task list

The **Task List** window lists all current embOS tasks. It retrieves its information directly from the embOS task list.



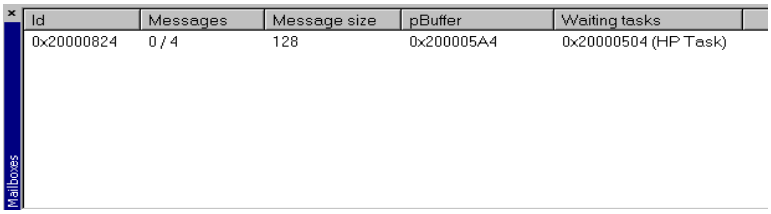
The individual columns are described below:

Column	Description
*	A green arrow points at the currently active embOS task.
Id	The task control block address that uniquely identifies a task.
Name	If available, the task name is shown here.
Status	The task status as a short text.
Timeout	If a task is delayed, this column shows the timeout value and in parentheses the point in time when the delay will be finished.
Events	The event mask of a task.
Stack Info	If available, this column shows the amount of used stack space, and the available stack space, as well as the value of the current stack bottom pointer.
Activations	The number of task activations.
Round Robin	If round robin scheduling is available, this column shows the number of remaining time slices and the number of time slice reloads.

Table 3.1: Task list window items

3.3 Mailboxes

A mailbox is a buffer that is managed by the real-time operating system. The buffer behaves like a normal buffer; you can put something (called a message) in and retrieve it later. This window shows the mailboxes and provides information about the number of messages, waiting tasks etc.

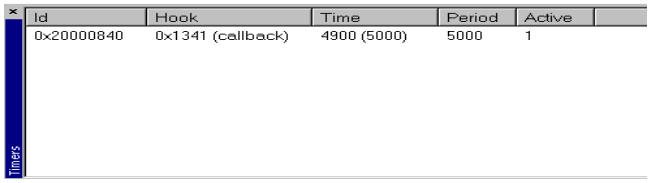


Column	Description
Id	The mailbox address.
Messages	The number of messages in a mailbox and the maximum number of messages as mailbox can hold.
Message size	The size of an individual message in bytes.
pBuffer	The message buffer address.
Waiting tasks	The list of tasks that are waiting for a mailbox, that is their address and name.

Table 3.2: Mailboxes window items

3.4 Timers

A software timer is an object that calls a user-specified routine after a specified delay. This window provides information about active software timers.

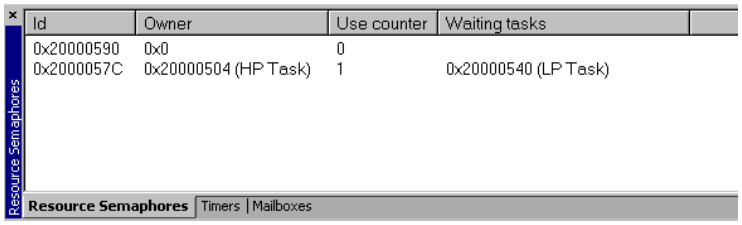


Column	Description
Id	The timer's address.
Hook	The function (address and name) that is called after the timeout.
Time	The time delay and the point in time, when the timer finishes waiting.
Period	The time period the timer runs.
Active	Shows whether the timer is active or not.

Table 3.3: Timers window items

3.5 Resource semaphores

Resource semaphores are used to manage resources by avoiding conflicts caused by simultaneous use of a resource. This window provides information about available resources.

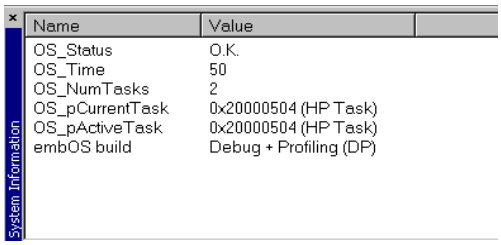


Column	Description
Id	The resource semaphore address.
Owner	The address and name of the owner task.
Use counter	Counts the number of semaphore uses.
Waiting tasks	Lists the tasks (address and name) that are waiting at the semaphore.

Table 3.4: Resource Semaphores window items

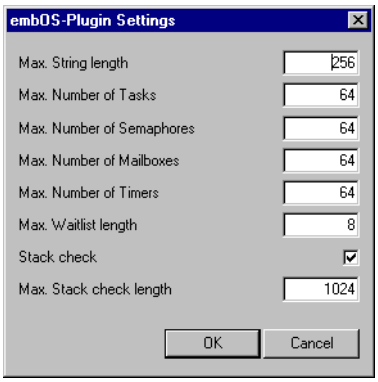
3.6 System information

A running embOS contains a number of system variables that are available for inspection. This window lists the most important ones.



3.7 Settings

To be safe, the embOS C-Spy plug-in imposes certain limits on the amount of information retrieved from the target, to avoid endless requests in case of false values in the target memory. This dialog box allows you to tweak these limits in a certain range, for example if your task names are no longer than 32 characters you may set the **Maximum string length** to 32, or if they are longer than the default you may increase that value.



After changing settings and clicking the **OK** button, your changes are applied immediately and should become noticeable after the next window update, for example when hitting the next breakpoint. However, the settings are restored to their default values on plug-in reload.

3.8 About

Finally, the **About** dialog box contains the embOS C-Spy plug-in version number and the date of compilation.

Chapter 4

Support

This chapter contains information about contacting support and what information to provide.

4.1 Contacting Support

We work hard to avoid as much software defects as possible. However, if you encounter an error in our software, you may contact our support at `support@segger.com`. We will try to correct any malfunction as soon as possible. To do this, we need all relevant information. Please try to provide us with at least the following information:

- IAR Embedded Workbench IDE & C-SPY debugger versions.
- Information about the target CPU.
- embOS C-Spy plug-in version number.
- A detailled description of the problem and how to reproduce it.
- If possible send us a project that triggers the problem.

