embOS

Real-Time Operating System

embOS plug-in for IAR C-Spy Debugger

Document: UM01025
Software Version: 3.5
Revision: 0
Date: June 3, 2020

A product of SEGGER Microcontroller GmbH

www.segger.com

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 GmbH (SEGGER) assumes no responsibility for any errors or omissions. SEGGER makes and you receive no warranties or conditions, express, implied, statutory or in any communication with you. SEGGER 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 GmbH

Ecolab-Allee 5
D-40789 Monheim am Rhein

Germany

Tel.	+49 2173-99312-0
Fax.	+49 2173-99312-28
E-mail:	support@segger.com*
Internet:	www.segger.com

*By sending us an email your (personal) data will automatically be processed. For further information please refer to our privacy policy which is available at https://www.segger.com/legal/privacy-policy/.

Manual versions

This manual describes the current software version. If you find an error in the manual or a problem in the software, please inform us and we will try to assist you as soon as possible. Contact us for further information on topics or functions that are not yet documented.

Print date: June 3, 2020

Software	Revision	Date	By	Description
3.5	0	200603	MC	New plug-in versions 6.10.3.5, 7.10.3.5, 7.50.3.5, 8.10.3.5, and 8.30.3.5.
3.4	0	200325	MM	New plug-in versions 6.10.3.4, 7.10.3.4, 7.50.3.4, 8.10.3.4, and 8.30.3.4.
3.3	0	191205	MC	New plug-in versions 8.30.3.3.
3.2	0	191106	MC	New plug-in versions 6.10.3.2, 7.10.3.2, 7.50.3.2, 8.10.3.2, and 8.30.3.2.
3.1	1	180924	MC	Updated to include most recent versions of IAR embedded workbench and their compatible plug-ins.
3.1	0	180503	MC	New plug-in versions 6.10.3.1, 7.10.3.1, 7.50.3.1, 8.10.3.1, and 8.30.3.1.
3.0	0	170915	MM	New plug-in versions 6.10.3.0, 7.10.3.0, 7.50.3.0, and 8.10.3.0.

About this document

Assumptions

This document assumes that you already have a solid knowledge of the following:

- The software tools used for building your application (assembler, linker, C compiler).
- The C programming language.
- The target processor.
- DOS command line.

If you feel that your knowledge of C is not sufficient, we recommend *The C Programming Language* by Kernighan and Richie (ISBN 0--13--1103628), which describes the standard in C programming and, in newer editions, also covers the ANSI C standard.

How to use this manual

This manual explains all the functions and macros that the product offers. It assumes you have a working knowledge of the C language. Knowledge of assembly programming is not required.

Typographic conventions for syntax

This manual uses the following typographic conventions:

Style	Used for
Body	Body text.
`Keyword`	Text that you enter at the command prompt or that appears on the display (that is system functions, file- or pathnames).
`Parameter`	Parameters in API functions.
`Sample`	Sample code in program examples.
`Sample comment`	Comments in program examples.
Reference	Reference to chapters, sections, tables and figures or other documents.
GUIElement	Buttons, dialog boxes, menu names, menu commands.
Emphasis	Very important sections.

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 debug sessions. This enables you to inspect the state of several embOS primitives such as the task list, queues, counting semaphores, resource semaphores, mailboxes, software timers, memory pools, event objects, watchdogs and major system variables.

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 Supported CPUs

The embOS C-Spy plug-in works with 8-bit, 16-bit or 32-bit CPUs in little- or big-endian mode. To use the embOS C-Spy plug-in you need a version of IAR Embedded Workbench installed and a debug target which uses embOS.

The following plug-ins are available and may be used with the listed versions of IAR's Embedded Workbench:

embOS Port	IAR Embedded Workbench Version	Compatible Plug-In version
78K0	≤ 4.80 ≥ 4.81	6.10.3.5 7.10.3.5
8051	≤ 8.30 ≥ 9.10 and ≤ 9.30 ≥ 10.10	6.10.3.5 7.10.3.5 8.10.3.5
ARM7 / ARM9 Cortex-A/R/M	≤ 6.70 ≥ 7.10 and ≤ 7.40 ≥ 7.50 and ≤ 7.80 ≥ 8.10 and ≤ 8.22 ≥ 8.30	6.10.3.5 7.10.3.5 7.50.3.5 8.10.3.5 8.30.3.5
AVR	≤ 6.40 ≥ 6.50 and ≤ 6.80 ≥ 7.10	6.10.3.5 7.10.3.5 8.10.3.5
AVR32	≤ 4.21 ≥ 4.30	6.10.3.5 7.10.3.5
Coldfire	Any	3.82.3.0
CR16C	≤ 3.20 ≥ 3.30	6.10.3.5 7.10.3.5
H8	Any	6.0.1.0
M16C	≤ 3.60 ≥ 3.70	6.10.3.5 7.10.3.5
M32C	Any	6.0.1.0
MSP430	≤ 5.60 ≥ 6.10 and ≤ 6.50 ≥ 7.10	6.10.3.5 7.10.3.5 8.10.3.5
R32C	Any	6.10.3.5
RH850	≤ 1.30 = 1.40 = 2.10 ≥ 2.20	7.10.3.5 7.50.3.5 8.10.3.5 8.30.3.5
RL78	≤ 1.30 ≥ 1.40 and ≤ 2.21 = 3.10 ≥ 4.10	6.10.3.5 7.10.3.5 8.10.3.5 8.30.3.5
RX	≤ 2.50 ≥ 2.60 and ≤ 2.90 = 3.10 ≥ 4.10	6.10.3.5 7.10.3.5 8.10.3.5 8.30.3.5
SH	Any	6.10.3.5
STM8	≤ 1.42 ≥ 2.10 and ≤ 2.20 = 3.10 ≥ 3.11	6.10.3.5 7.10.3.5 8.30.3.5 8.10.3.5 8.30.3.5

embOS Port	IAR Embedded Workbench Version	Compatible Plug-In version
V850	≤ 4.10 ≥ 4.20	6.10.3.5 7.10.3.5

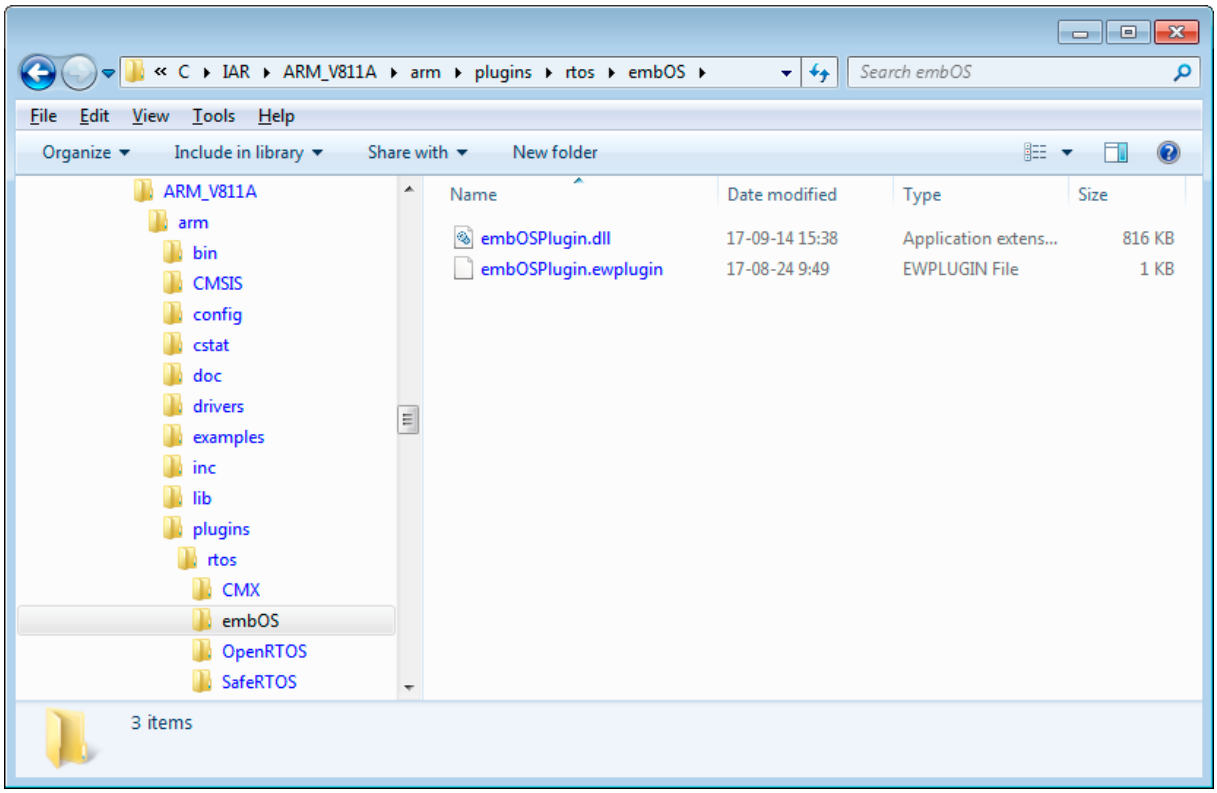
Chapter 2

Installation

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

2.1 Installation Procedure

Typically, there is no installation required since the IAR Embedded Workbench comes with the plug-in already pre-installed. In case you want to update the plug-in to a more recent version, however, you would need to replace two files that are located within the Embedded Workbench installation directory with the respective files from the embOS C-Spy plug-in package. The Embedded Workbench installation directory should resemble the following structure:

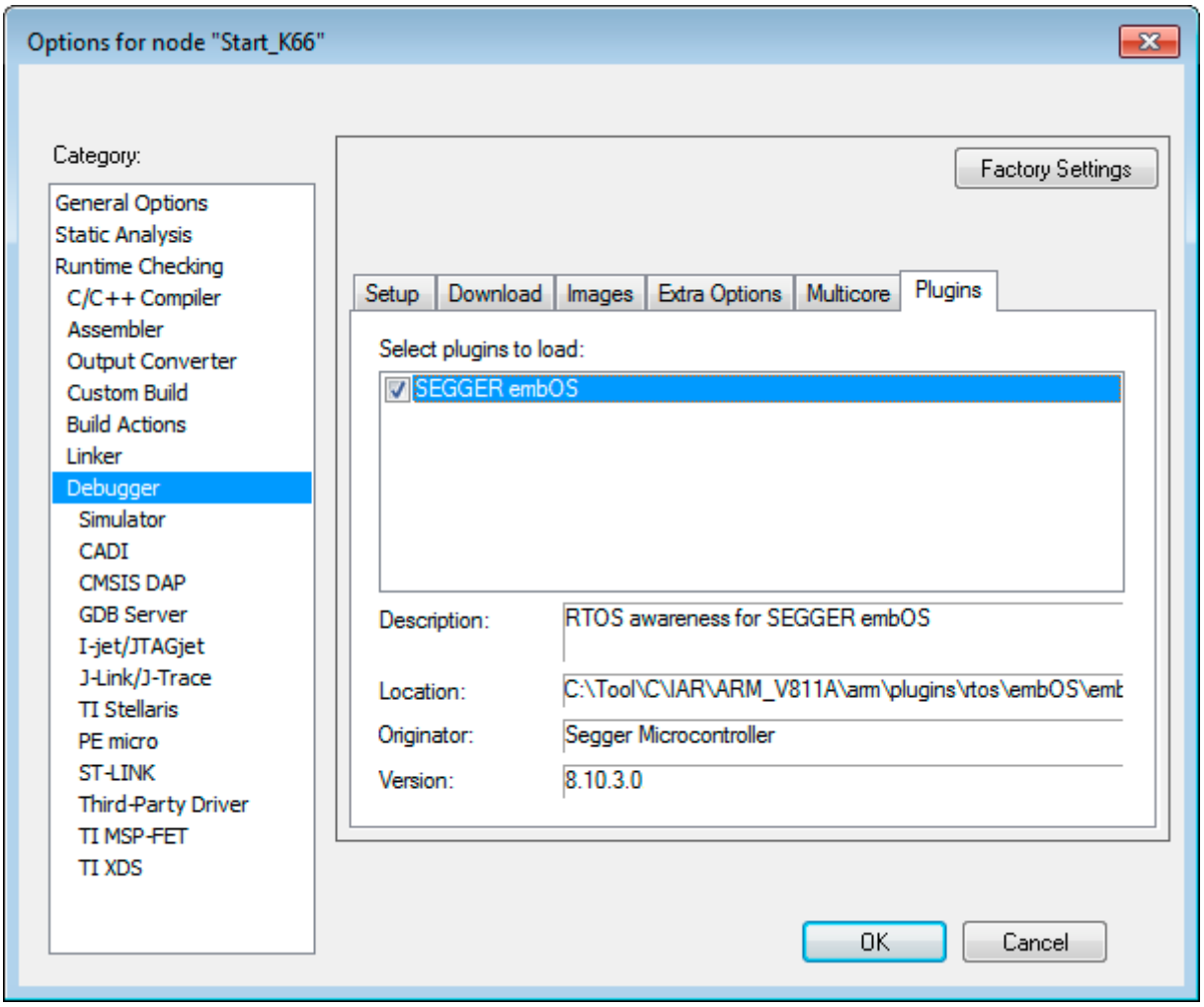


If appropriate folders do not yet exist with your installation, you should create a directory called `embOS` within the CPU specific folder `plugin\rtos\`, and subsequently copy the files from the embOS C-Spy plug-in package into that folder. Note that IAR Embedded Workbench must not be running during the update process.

> **Note**
>
> Before replacing any files already found in the `plugin\rtos\embOS` folder of the IAR Embedded Workbench, you may want to backup these files. You should also check the version number of the plug-in inside `embOSPlugin.ewplugin`. Therein, the version number is shown as the last entry and looks like follows:
>
> `<version>8.30.3.2</version>`
>
> The first part, `8.30`, is the major version number and indicates the C-Spy SDK this plug-in was created with. Typically, it is not recommended to replace previous plug-in versions with more recent major versions, but with more recent minor versions only. Minor versions are indicated by the second part of the version number, i.e. `3.2`. It's recommended to replace the plug-in currently installed with your Embedded Workbench if its minor version is lower than the minor version of the plug-in that is shipped with embOS.

2.2 Configuration

By default, embOS start projects will enable the embOS C-Spy plug-in upon project load. The plug-in may be explicitly disabled, individually for each project configuration, in the debugger section of the project's options:



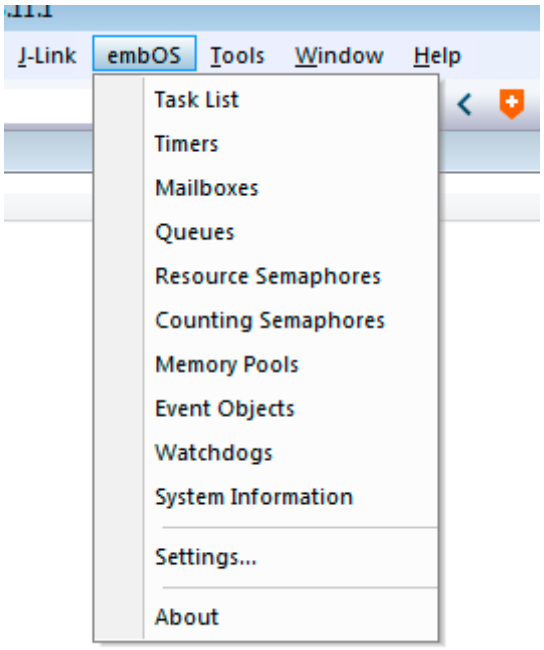
Chapter 3

Getting Started

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

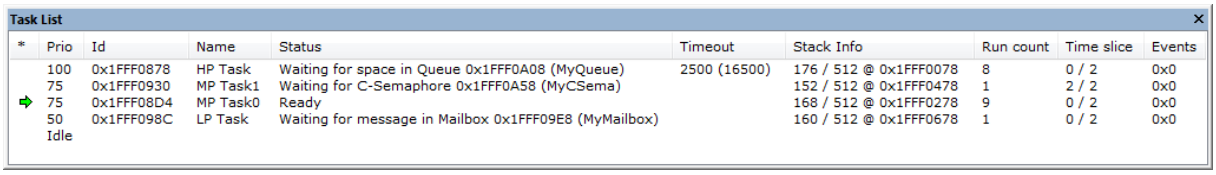
3.1 Overview

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. A Release build, for instance, won't show any information about semaphores, queues, event objects, or mailboxes.

3.2 Task List



Column	Description
*****	A green arrow points at the currently active embOS task.
Prio	The priority of the 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. The name in the parentheses belongs to the object (semaphore, mailbox, etc.) the task is waiting for.
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.
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.
Run count	The number of task activations.
Time slice	If round robin scheduling is available, this column shows the number of remaining time slices and the number of time slice reloads.
Events	The event mask of a task.

3.2.1 Task sensitivity

The Source Code window, the Disassembly window, the Register window, and the Call Stack window of the C-Spy debugger are task sensitive since version 3.62 of the embOS C-Spy plug-in for several CPUs. This means that they show the position in the code, the general-purpose registers and the call stack of the selected task. By default, the selected task is always the running task, which is the normal behavior of a debugger that the user expects.

You can examine a particular thread by double-clicking on the corresponding row in the window. The selected task will be underlayed in yellow. The C-Spy Debugger rebuilds the call stack and the preserved general-purpose registers of a suspended task. Refer to *State of suspended tasks* on page 18 for detailed information about which information are available for the different task states.

Every time the CPU is started or when the Idle-row of the task window is double clicked, the selected task is switched back to this default.

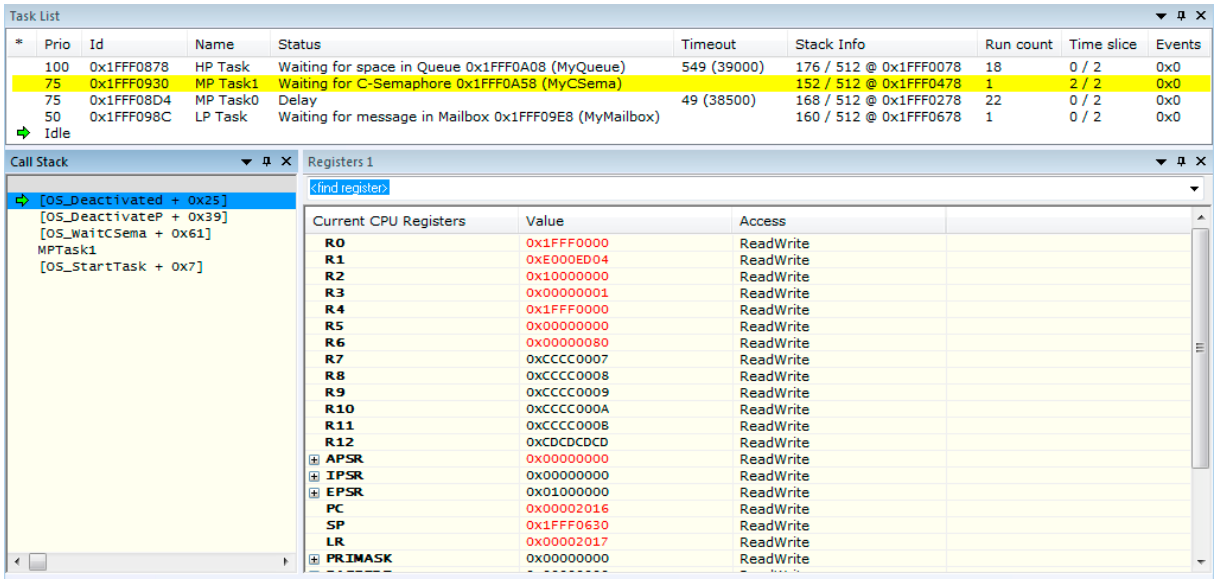
The task sensitive source window, call stack and register window are supported for the following CPUs:

- ARM7 / ARM9
- ARM Cortex-A
- ARM Cortex-R
- ARM Cortex-M
- Renesas M16C
- Renesas R32C
- Renesas RL78
- Renesas RX
- Renesas SH2A

3.2.1.1 State of suspended tasks

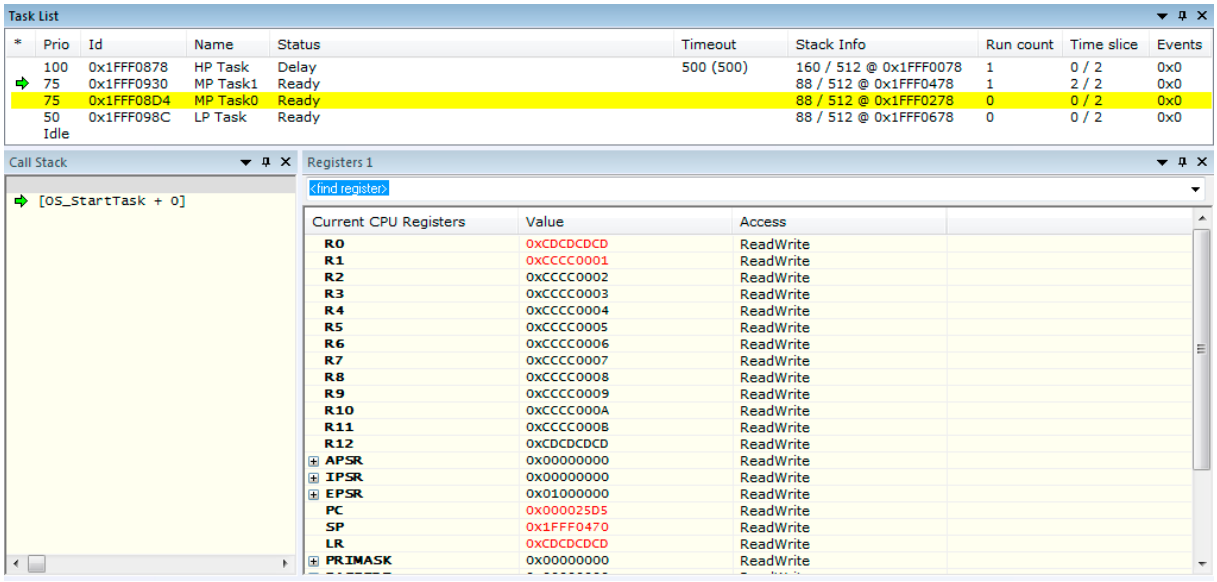
Blocked tasks

Tasks which have given up execution voluntarily by calling a blocking function, such as `OS_Delay()` or `OS_Wait_…()`. In this case, the OS saved all registers which can be viewed in the Register window. It might be that scratch registers are not saved and thus not displayed in the Register window.

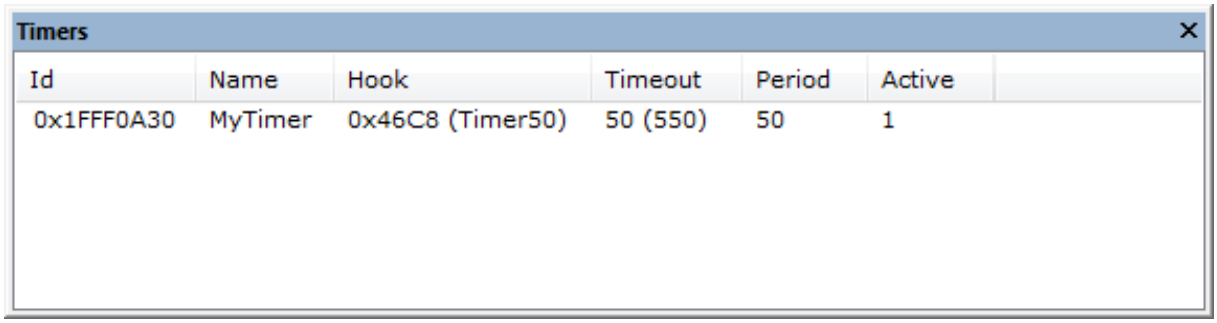


Tasks waiting for first activation

These basically fall into the same category as blocked tasks, the call stack and registers look similar to the following screenshots. Similarly, temporary registers are unknown. The `Call Stack` shows a single entry `OS_StartTask`. Run count is 0.



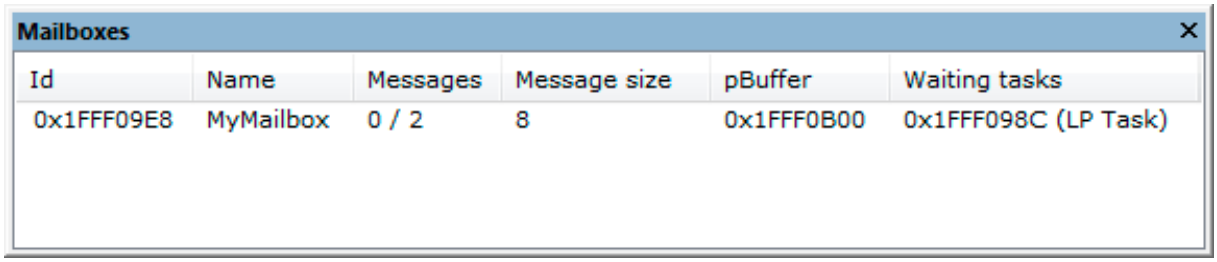
3.3 Timers



Column	Description
Id	The timer’s address.
Name	If available, the respective object identifier is shown here.
Hook	The function (address and name) that is called after the timeout.
Timeout	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 (running) or not.

3.4 Mailboxes

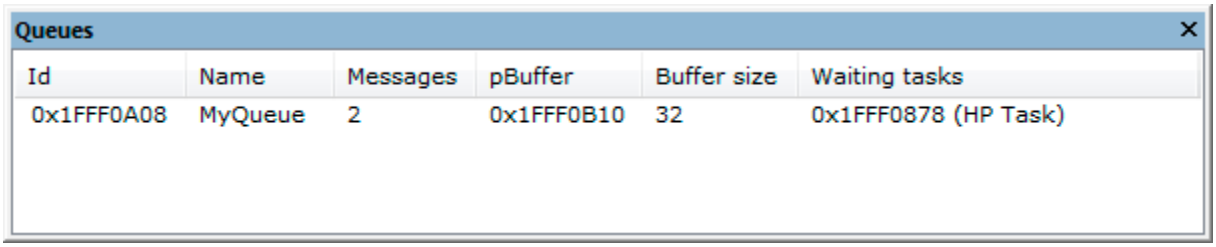
This view displays information in debug builds of embOS only.



Column	Description
Id	The mailbox address.
Name	If available, the respective object identifier is shown here.
Messages	The number of messages in a mailbox and the maximum number of messages the 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 the mailbox (address and, if available, name). Only those tasks that are displayed in the task list window are shown here.

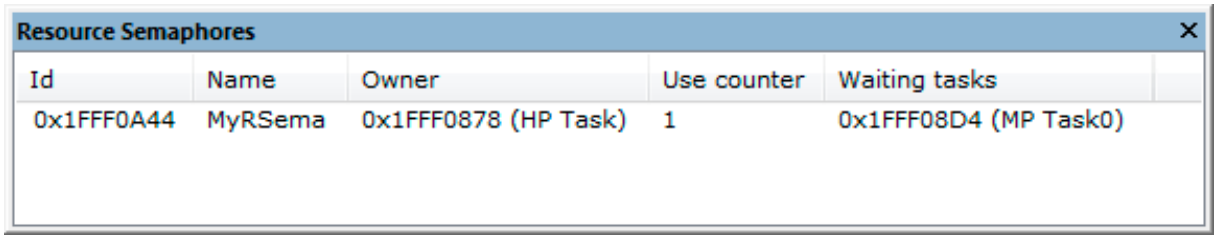
3.5 Queues

With embOS V4.38 and subsequent versions, this view displays information in debug builds of embOS only.



Column	Description
Id	The queue address.
Name	If available, the respective object identifier is shown here.
Messages	The number of messages in a queue.
pBuffer	Address of the buffer which contains the messages.
Buffer size	The size of the message buffer.
Waiting tasks	The list of tasks that are waiting for the queue (address and, if available, name). Only those tasks that are displayed in the task list window are shown here.

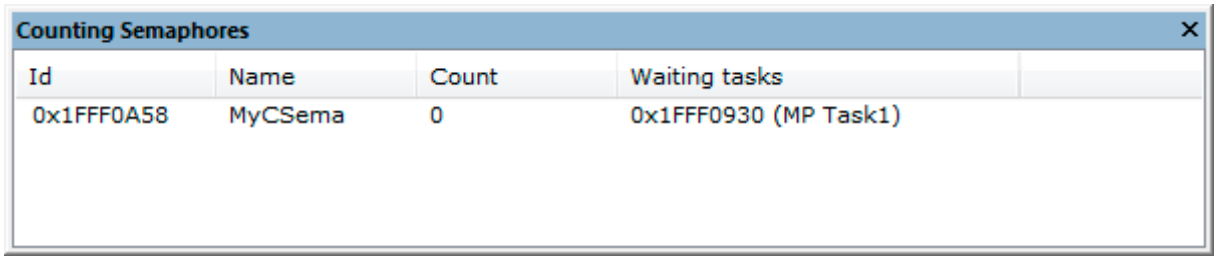
3.6 Resource Semaphores



Column	Description
Id	The resource semaphore address.
Name	If available, the respective object identifier is shown here.
Owner	The address and name of the owner task.
Use counter	Counts the number of semaphore uses.
Waiting tasks	The list of tasks that are waiting for the semaphore (address and, if available, name). Only those tasks that are displayed in the task list window are shown here.

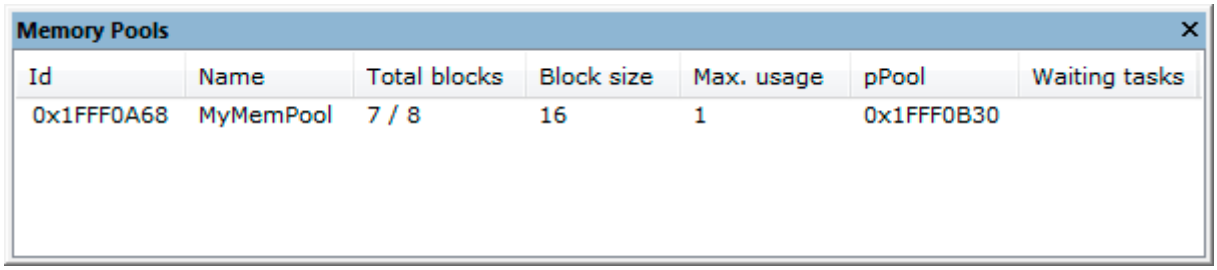
3.7 Counting Semaphores

This view displays information in debug builds of embOS only.



Column	Description
Id	The counting semaphore address.
Name	If available, the respective object identifier is shown here.
Count	Counts how often this semaphore can be claimed until it blocks.
Waiting tasks	The list of tasks that are waiting for the semaphore (address and, if available, name). Only those tasks that are displayed in the task list window are shown here.

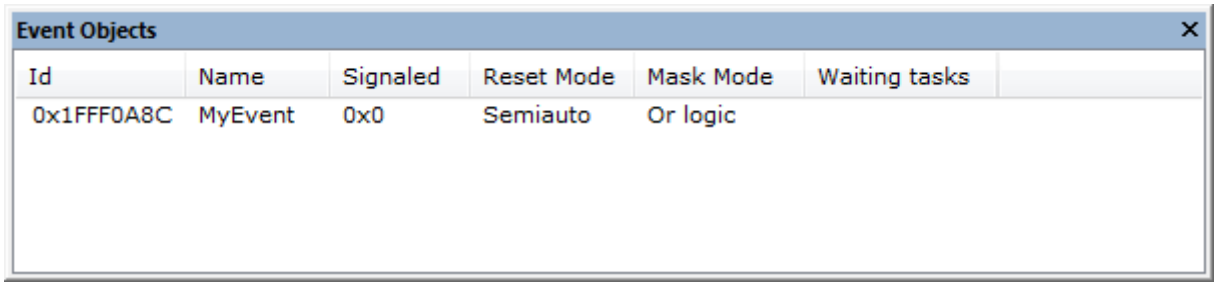
3.8 Memory Pools



Column	Description
Id	The memory pool address.
Name	If available, the respective object identifier is shown here.
Total blocks	Shows the available blocks and the maximal number of blocks.
Block size	Shows the size of a single memory block.
Max. usage	Shows the maximal count of blocks which were simultaneously allocated.
pPool	The address of the memory pool buffer.
Waiting tasks	The list of tasks that are waiting for free blocks in the memory pool (address and, if available, name). Only those tasks that are displayed in the task list window are shown here.

3.9 Event Objects

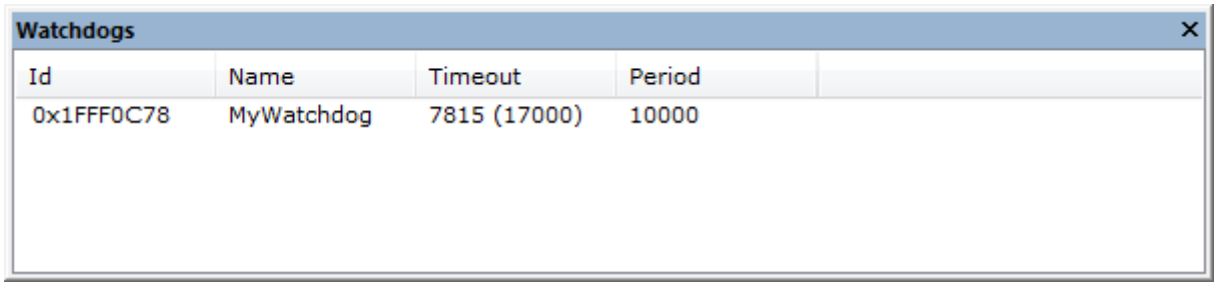
This view displays information in debug builds of embOS only. This view displays information with embOS V4.38 and subsequent versions only.



Column	Description
Id	The event object address.
Name	If available, the respective object identifier is shown here.
Signaled	The hexadecimal value of the bit mask containing the signaled event bits.
Reset Mode	The event objects reset mode.
Mask Mode	The current mask mode indicating whether Or or And logic is used to check whether a task shall resume.
Waiting tasks	The list of tasks that are waiting for an event object (address and, if available, name). Only those tasks that are displayed in the task list window are shown here.

3.10 Watchdogs

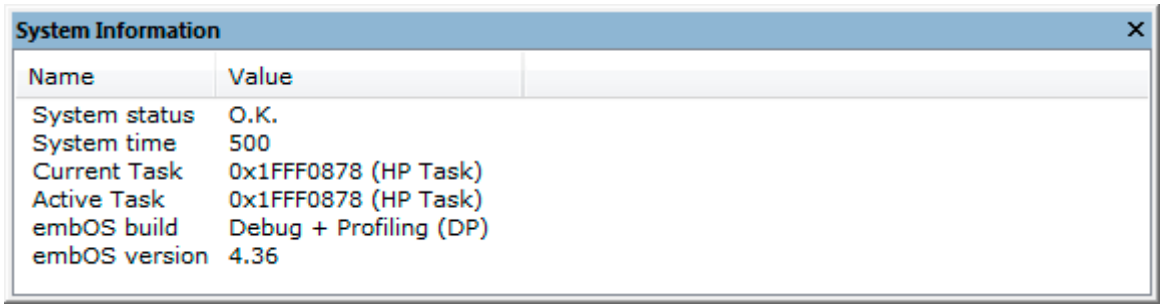
This view displays information with embOS V4.38 and subsequent versions only.



Column	Description
Id	The watchdog address.
Name	If available, the respective object identifier is shown here.
Timeout	The remaining time (and the system time in parentheses) until the watchdog has to be fed.
Period	The period in which the watchdog has to be fed.

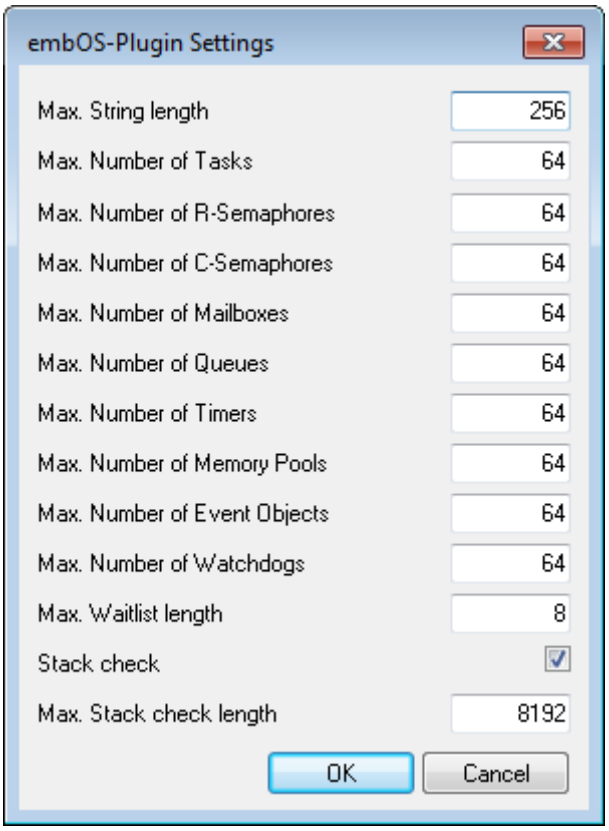
3.11 System Information

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



3.12 Settings

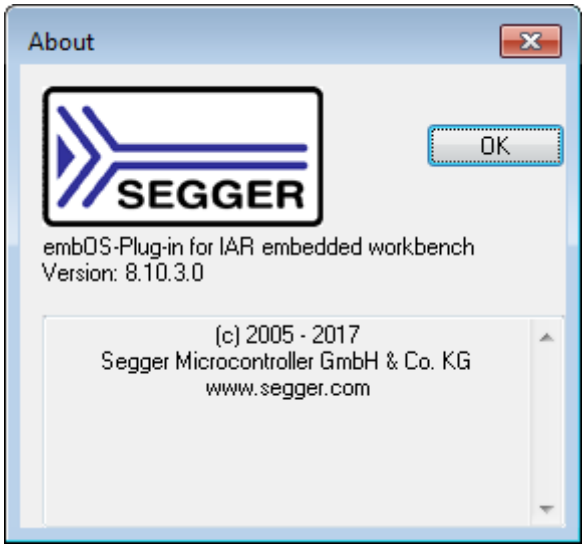
in order to avoid endless requests in case of erroneous data in the target memory, the embOS C-Spy plug-in imposes certain limits on the amount of information retrieved from the target. This dialog box allows you to tweak these limits within a certain range. For example, if your task names are no longer than 32 characters, you might want to configure the `Max. string length` to 32. Alternatively, if task names are longer than the default value, you may increase that value according to your needs.



When clicking the `OK` button, changes to the settings are applied immediately and become visible in the plug-in windows upon the next window update, e.g. when hitting a breakpoint. However, the settings are restored to their default values on plug-in reload.

3.13 About

The About dialog box contains the embOS C-Spy plug-in version number.



Chapter 4

Support

4.1 Contacting support

This chapter should help if any problem occurs and describes how to contact the embOS support.

If you are a registered embOS user there are different ways to contact the embOS support:
1. You can create a support ticket via email to ticket_embos@segger.com.*
2. You can create a support ticket at segger.com/ticket.*
3. You can send an email to support_embos@segger.com.*

Please include the following information in the email or ticket:

- Which embOS do you use? (CPU, compiler).
- The embOS version.
- Your embOS registration number.
- If you are unsure about the above information you can also use the name of the embOS zip file (which contains the above information).
- A detailed description of the problem.
- Optionally a project with which we can reproduce the problem.

> **Note**
>
> Even without a valid license, feel free to contact our support e.g. in case of questions during your evaluation of embOS or for hobbyist purposes.

Please also take a few moments to help us improve our services by providing a short feedback once your support case has been solved.

*By sending us an email your (personal) data will automatically be processed. For further information please refer to our privacy policy which is available at https://www.segger.com/legal/privacy-policy/.

