

# ***Getting started with SEGGER Eval Software***

Document revision 0

Date: April 1, 2008



A product of SEGGER Microcontroller GmbH & Co. KG

**[www.segger.com](http://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 & Co. KG (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

You may not extract portions of this manual or modify the PDF file in any way without the prior written permission of the manufacturer. The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such a license.

© 2008 SEGGER Microcontroller GmbH & Co. KG, Hilden / Germany

## 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 & Co. KG

Heinrich-Hertz-Str. 5  
D-40721 Hilden

Germany

Tel. +49 2103-2878-0

Fax. +49 2103-2878-28

Email: [support@segger.com](mailto:support@segger.com)

Internet: <http://www.segger.com>

## Manual versions

This manual describes the latest software version. If any error occurs, please inform us and we will try to assist you as soon as possible.

For further information on topics or routines not yet specified, please contact us.

Manual version	Date	By	Explanation
1.00	080328	SK	Initial version.

# 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.
Reference	Reference to chapters, sections, tables and figures or other documents.
GUI Element	Buttons, dialog boxes, menu names, menu commands.
Emphasis	Very important sections

**Table 1.1: Typographic conventions**



**SEGGER Microcontroller GmbH & Co. KG** develops and distributes software development tools and ANSI C software components (middleware) for embedded systems in several industries such as telecom, medical technology, consumer electronics, automotive industry and industrial automation.

SEGGER's intention is to cut software development-time for embedded applications by offering compact flexible and easy to use middleware, allowing developers to concentrate on their application.

Our most popular products are emWin, a universal graphic software package for embedded applications, and embOS, a small yet efficient real-time kernel. emWin, written entirely in ANSI C, can easily be used on any CPU and most any display. It is complemented by the available PC tools: Bitmap Converter, Font Converter, Simulator and Viewer. embOS supports most 8/16/32-bit CPUs. Its small memory footprint makes it suitable for single-chip applications.

Apart from its main focus on software tools, SEGGER develops and produces programming tools for flash microcontrollers, as well as J-Link, a JTAG emulator to assist in development, debugging and production, which has rapidly become the industry standard for debug access to ARM cores.

#### Corporate Office:

<http://www.segger.com>

#### United States Office:

<http://www.segger-us.com>

## EMBEDDED SOFTWARE (Middleware)



### emWin

#### Graphics software and GUI

emWin is designed to provide an efficient, processor- and display controller-independent graphical user interface (GUI) for any application that operates with a graphical display. Starterkits, eval- and trial-versions are available.



### embOS

#### Real Time Operating System

embOS is an RTOS designed to offer the benefits of a complete multitasking system for hard real time applications with minimal resources. The profiling PC tool embOSView is included.



### emFile

#### File system

emFile is an embedded file system with FAT12, FAT16 and FAT32 support. emFile has been optimized for minimum memory consumption in RAM and ROM while maintaining high speed. Various Device drivers, e.g. for NAND and NOR flashes, SD/MMC and CompactFlash cards, are available.



### USB-Stack

#### USB device stack

A USB stack designed to work on any embedded system with a USB client controller. Bulk communication and most standard device classes are supported.

## SEGGER TOOLS

### Flasher

#### Flash programmer

Flash Programming tool primarily for microcontrollers.

### J-Link

#### JTAG emulator for ARM cores

USB driven JTAG interface for ARM cores.

### J-Trace

#### JTAG emulator with trace

USB driven JTAG interface for ARM cores with Trace memory. supporting the ARM ETM (Embedded Trace Macrocell).

### J-Link / J-Trace Related Software

Add-on software to be used with SEGGER's industry standard JTAG emulator, this includes flash programming software and flash breakpoints.



# Table of Contents

1	Introduction .....	7
1.1	Setup.....	8
1.2	Project structure.....	9
1.3	Eval limitations.....	10
2	Getting started .....	11
2.1	Running a sample application .....	12



# Chapter 1

## Introduction

---

# 1.1 Setup

## Requirements

In order to recompile the projects you will need the IAR Embedded Workbench as indicated in the ReadMe.txt file.

## Installation

All eval packages are supplied as a zip-file. Extract it to any folder of your choice, preserving the directory structure of the zip-file. Assuming that you are using the IAR Embedded Workbench project manager to develop your application, no further installation steps are required. You will find prepared sample start applications, which you should use and modify to write your application.

All eval packages include evaluation builds some or all of the following products:

Component	Description
emFile	SEGGER's file system for embedded applications.
embOS	SEGGER's priority-controlled real time operating system.
emOS/IP	SEGGER's CPU independent TCP/IP stack.
emUSB	SEGGER's high speed USB device stack.

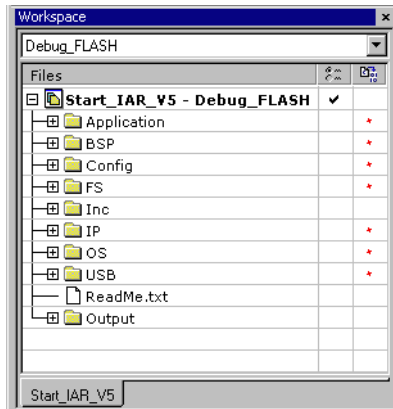
**Table 1.1: Eval package components**

All products can be combined and purchased separately. For ordering information, please contact SEGGER, [www.segger.com](http://www.segger.com).



## 1.2 Project structure

All components of the eval package are provided as libraries and the required header files. The sample applications are provide in source form. The sample projects are organized in the following way:



The root directory of the eval package includes two additional directories (`Doc` and `Windows`), which are not included in the project tree of the IAR Embedded Workbench project.

Directory	Description
Application	Includes the sample applications.
BSP	Includes the Board Support Package (BSP). The BSP consists all files which are required to initialize the hardware and build a project.
Config	Includes the configuration files of the included software packages.
Doc	Includes the user guides to all provided products and the <code>License.txt</code> .
FS	Includes the emFile header files and the emFile libraries.
Inc	Includes utility header files which are not directly related to one of the included software packages.
IP	Includes the embOS/IP header files and the embOS/IP libraries.
OS	Includes the embOS header files and the embOS libraries and <code>main.c</code> .
USB	Includes the header files and the emUSB libraries.
Windows	Includes the Microsoft Windows examples and the required drivers for the USB components BULK communication and Communication Device Class (CDC). Refer to the <i>emUSB User Guide</i> for more information about the installation of these drivers.

**Table 1.2: Eval package structure**

## 1.3 Eval limitations

The included eval versions of the different components of the eval package have the following limitations:

Component	Description
emFile	The eval version of the emFile libraries can only handle one open file at any given time.
embOS	The eval version of the embOS libraries run without a time limit with a maximum of three tasks. If your application creates more than three tasks stops embOS after a time limit of 15 minutes.
emOS/IP	The eval version of the embOS/IP libraries have a time limit of 15 minutes on the connection.
emUSB	The eval version of the emUSB libraries have a time limit of 15 minutes on the connection.

**Table 1.3: Limitations of eval package components**

Your use of the eval package or of any part included in the project indicates your acknowledgment and agreement to the SEGGER eval software license . `License.txt` is located in the `Doc` directory of the eval package.

# Chapter 2

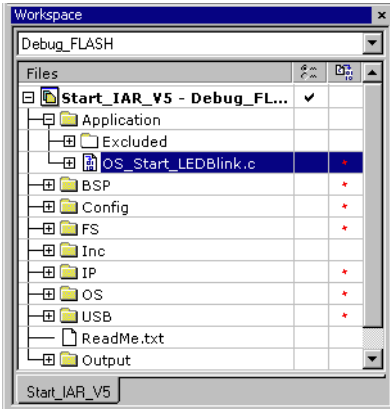
## Getting started

---

This chapter contains all required information to start working with the sample applications.

## 2.1 Running a sample application

The application folder includes the selected sample application, normally `OS_Start_LEDBlink.c` and a folder which is excluded from build. The **Excluded** folder contains all available sample applications. All available samples contain a function `MainTask()`. `MainTask()` is called from program main which is located in the **OS** folder.



`main.c` is required by all supplied sample applications.

```
#include "RTOS.H"
#include "BSP.h"

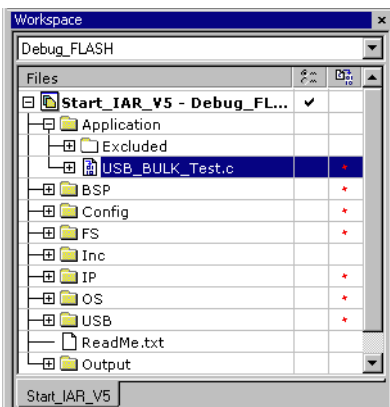
void MainTask(void);

static OS_STACKPTR int Stack0[512]; /* Task stacks */
static OS_TASK TCB0; /* Task-control-blocks */

/*****
 *
 *      main
 */
void main(void) {
    OS_IncDI(); /* Initially disable interrupts */
    OS_InitKern(); /* Initialize OS */
    OS_InitHW(); /* Initialize Hardware for OS */
    BSP_Init(); /* Initialize BSP module */
    BSP_SetLED(0);
    OS_CREATETASK(&TCB0, "MainTask", MainTask, 100, Stack0);
    OS_Start();
}
```

`main()` creates a single task called `MainTask`, which is implemented in every sample application.

Open the **Application** folder and use "drag and drop" in the **Workspace** window of the IAR Embedded Workbench to change the sample application. Move the included sample application (e.g. `OS_Start_LEDBlink.c`) in the **Excluded** folder and move the favored sample application from the **Excluded** folder to the **Application** folder.



Some embOS/IP and emUSB sample applications are client/server applications and consist of an embedded side and a PC side. The PC applications are provided in source code and as precompiled executable (.exe). For compiling the example application you need a Microsoft compiler. The compiler is part of Microsoft Visual C++ 6.0 or Microsoft Visual Studio .Net. All Windows host sample applications are located in the Windows folder of your eval project.

