SEGGER
Eval Software

Getting Started with
SEGGER Eval Software

Document: AN00020

Revision: 15

Date: January 4, 2018

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 (SEG-GER) 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

You may not extract portions of this manual or modify the PDF file in any way without the prior written permission of SEGGER. 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.

© 2014-2018 SEGGER Microcontroller GmbH, 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

In den Weiden 11
D-40721 Hilden

Germany

Tel.	+49 2103-2878-0
Fax.	+49 2103-2878-28
E-mail:	support@segger.com
Internet:	*www.segger.com*

3

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: January 4, 2018

Revision	Date	By	Description
15	180103	NG	Removed chapter "Licensing" Removed section "Eval limitations" Added chapter "Appendix" Moved sections "Project structure" and "Software components in the package" to chapter "Appendix" Renamed chapter "How to start" to "Getting started" Moved section "Setup" to chapter "Getting started" Updated section "Setup" Restructured chapter "Getting started"
14	171215	NG	Ported document to emDoc. Fixed various typos. Removed chapter "Sample applications".
13	170103	RH	New embOS/IP and emModbus sample names added to chapter "Sample applications".
12	161128	RH	Descriptions for emSSL and emCompress added, chapters "How to start", "Embedded Studio" and "Sample applications" updated.
11	160713	RH	Chapter "Sample applications" updated.
10	160415	RH	Minor issues fixed.
9	160121	RH	Chapter "How to start" updated, wording & spelling improved.
8	151130	RH	Chapter "SEGGER Embedded Studio" updated.
7	151119	RH	emModbus and new sample descriptions added.
6	150825	MC	Added the Keil MDK chapter.
5	150701	MC	Fixed erroneous descriptions, improved wording & spelling.
4	150611	MC	Added the emIDE and SES chapters.
3	150515	SC	Added the Microchip MPLAB X chapter.
2	150319	YR	Added the Renesas e2 studio chapter.
1	150316	SC	Fixed some minor issues.
0	141208	SC	Initial version.

4

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 *C: A Reference Manual* by Harbison and Steele (ISBN 0–13–089592X). This book provides a complete description of the C language, the run-time libraries, and a style of C programming that emphasizes correctness, portability, and maintainability.

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.
Parameter	Parameters in API functions.
Sample	Sample code in program examples.
Sample comment	Comments in program examples.
User Input	Text entered at the keyboard by a user in a session transcript.
Secret Input	Text entered at the keyboard by a user, but not echoed (e.g. password entry), in a session transcript.
Reference	Reference to chapters, sections, tables and figures.
Emphasis	Very important sections.
SEGGER home page	A hyperlink to an external document or web site.

Table of contents

Chapter 1

Introduction

This chapter provides basic information about the purpose of this manual and the entire software package, which consists of different middleware components.

1.1   About this manual

This document describes the different SEGGER Eval Software components and the sample applications that are delivered with them.

1.2   What is the purpose of this package?

SEGGER Eval Software is designed to provide customers and potential customers (you) with a complete and easy to use software package for the specified target hardware and several IDEs (e.g. SEGGER Embedded Studio).
It allows straightforward evaluation of the target hardware, the target compiler and SEGGER's middleware components.

SEGGER Eval Software packages can be retrieved from the following location:
SEGGER website: Evalboards

1.2.1   What are the components of the software package?

The SEGGER Eval Software components are provided in library form, whereas the sample applications are provided as source code.
In addition, most SEGGER Eval Software packages include a **Prebuild**-folder containing prebuild executables. These may simply be downloaded to the target hardware, allowing users to completely omit the compilation of the source code.

1.2.2   Can I recompile the supplied applications?

Yes, you may modify and recompile any of the application samples that are included in this software package. While an appropriate target compiler is required to do so, even an evaluation version of the compiler can be used in most cases. Please refer to `ReadMe.txt` for more information on the respective target compiler.

1.2.3   Can I write my own applications with this package?

Yes, you may also write your own applications. However, the purpose of this SEGGER Eval Software is to test the soft- and hardware for fitness. You must not use the resulting application in a commercial product.

Regarding developing and distributing applications for non-commercial (e.g. hobbyist) use, please refer to `License.txt`.

Chapter 2

Getting started

This chapter provides information on how to start working with the sample applications.

2.1   Setup

Requirements

Compilation of the provided application samples requires an appropriate target compiler (or eval version of the target compiler) as indicated in `ReadMe.txt`.

Furthermore, if emWin was included in the software package, it also contains a ready-to-go project for Microsoft Visual Studio 2010 that allows a recompilation of the emWin simulation. To use the project, either Microsoft Visual Studio 2010 or Microsoft Visual Studio .Net are required.

Installation

All SEGGER Eval Software packages are supplied as ZIP-files. The most recent version of the software can be retrieved from the following location:
SEGGER website: Evalboards

Extract the ZIP-file to a folder of your choice while retaining the folder structure of the ZIP-file. In general, no further installation steps are required. You may simply use the provided sample applications or start to modify them in order to write your application.

SEGGER Eval Software packages include object code builds of some or all of the following products:

Component	Description
emCompress	SEGGER's compression system
emCrypt	SEGGER's cryptographic algorithm library
emFile	SEGGER's file system
emModbus	SEGGER's Modbus stack
embOS	SEGGER's real time operating system
embOS/IP	SEGGER's TCP/IP stack
emSecure	SEGGER's digital signature suite
emSSH	SEGGER's SSH software library
emSSL	SEGGER's SSL software library
emUSB Device	SEGGER's USB Device stack
emUSB Host	SEGGER's USB Host stack
emWin	SEGGER's graphic library and GUI

Each component may be purchased separately and can be used in any combination.
For ordering information, please contact us: `info@segger.com`

For more detailed information on each middleware, please refer to *Software components in the package* or the respective product manual (see *Literature and references* on page 41).

Getting started IDE project file

Each SEGGER Eval Software contains a project file called "Start", which will be referred to as "the project file" in the remainder of this chapter.

2.2   Evaluating the SEGGERDEMO

The SEGGERDEMO is the sample application activated by default, so no changes need to be made in order to build and run it. In order to run any other sample, please refer to *Running a sample application* and the chapter for the respective IDE for instructions on how to change the sample application, and then follow the same steps listed below.

The following is a generic step by step description to fully evaluate the SEGGERDEMO application sample. All pictures in this chapter are exemplary; the actual display will differ when using a different IDE. For IDE-specific descriptions, please refer to the following chapters. Furthermore, the steps described below may differ from hardware to hardware, but, generally, will be similar.

2.2.1   Step 1: Open the project

- Get a fresh copy of the project and files.
- Open the project file with your IDE. Your screen should look similar to the screenshot below.
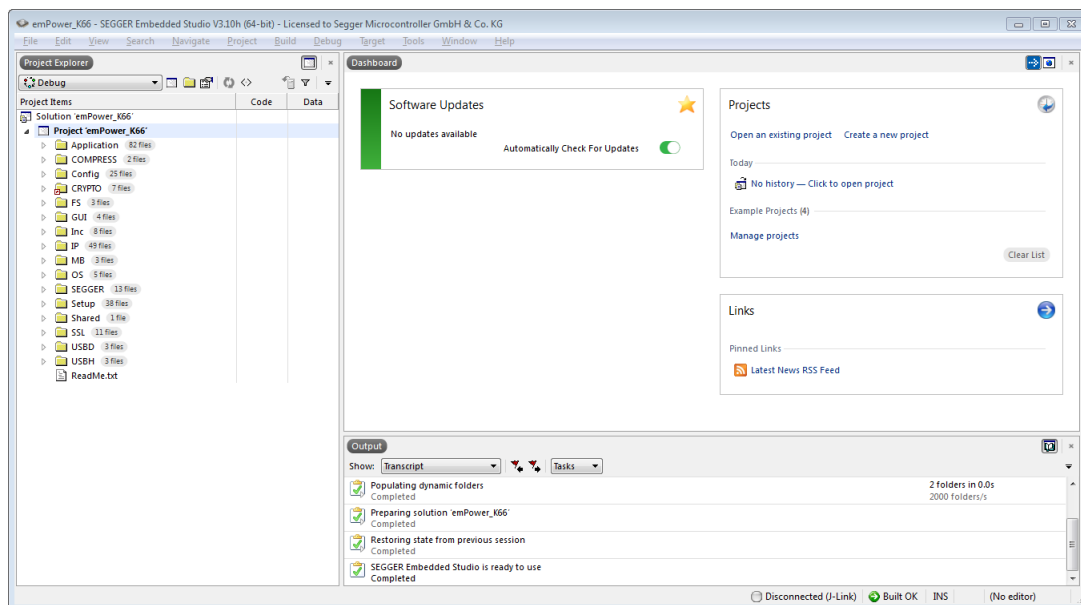
2.2.2   Step 2: Establish connections

- Make sure that your eval board is powered.
- Make sure your eval board and your PC are connected via your debug probe (e.g. SEGGER J-Link or an onboard debugger).

Optional:

- Connect your eval board with a RS232 cable to your PC.[1]
- Connect your target to a network with an Ethernet cable. Ensure that your PC is connected to the same network.[2]
- Establish an USB connection between your eval board and your PC via USB cable.[3]
- Plug in a Human Interface Device to your eval board.[4]

1.Only applies if UART is supported by the BSP
2.Only applies if the BSP contains embOS/IP
3.Only applies if the BSP contains emUSB Device
4.Only applies if the BSP contains emUSB Host

2.2.3   Step 3: Configuration

- If you are not using a DHCP server in your network you will need to manually configure the IP address and the subnet mask for your target in the file **Start\Setup \IP_Config_*.c**. The screenshot below shows an example configuration using the IP address 192.168.5.5 with subnet mask 255.255.0.0.
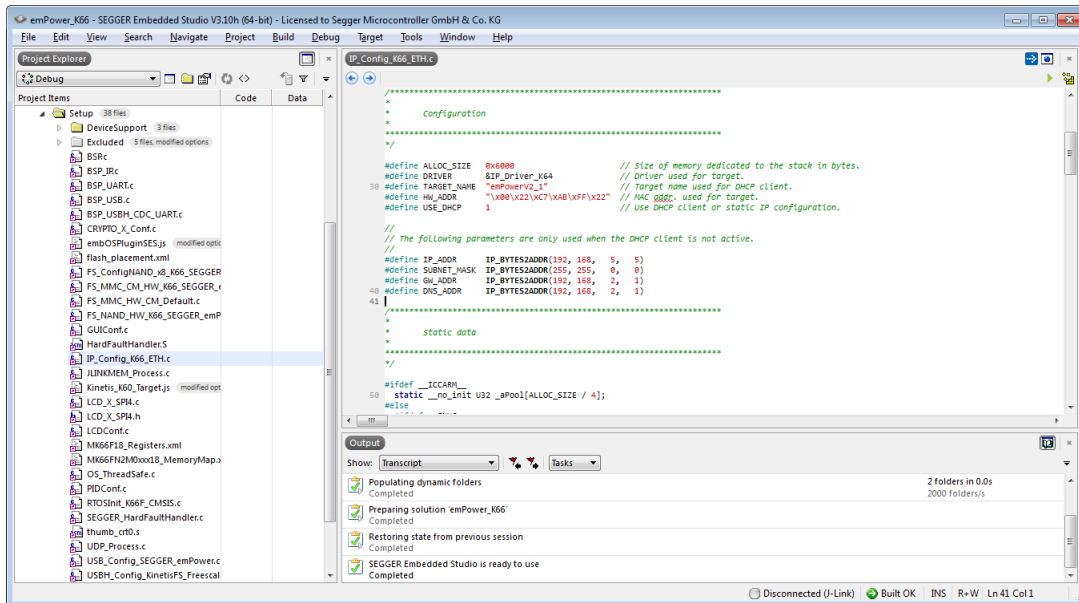

2.2.4   Step 4: Start the application

- Build the project. The build log should report no errors or warnings.
- Download the application and start the debug session. Your application should halt at main().
- Upon continuation of the application, the SEGGERDEMO should be displayed on the LCD.[1]

1.Only applies if the BSP contains emWin

2.2.5   Step 5: Evaluate the middleware components

- Start **Windows\OS\embOSView.exe** to verify that the UART is sending OS specific data.[1] Your screen should look similar to the screenshot below.
  Alternatively, if you are using the SEGGER J-Link, instead of communicating via UART you may also use the J-Link interface or Ethernet to communicate with embOSView. In case of deviant results, please refer to the embOS Generic Documentation (`Doc\UM01001_embOS.pdf`) for troubleshooting.
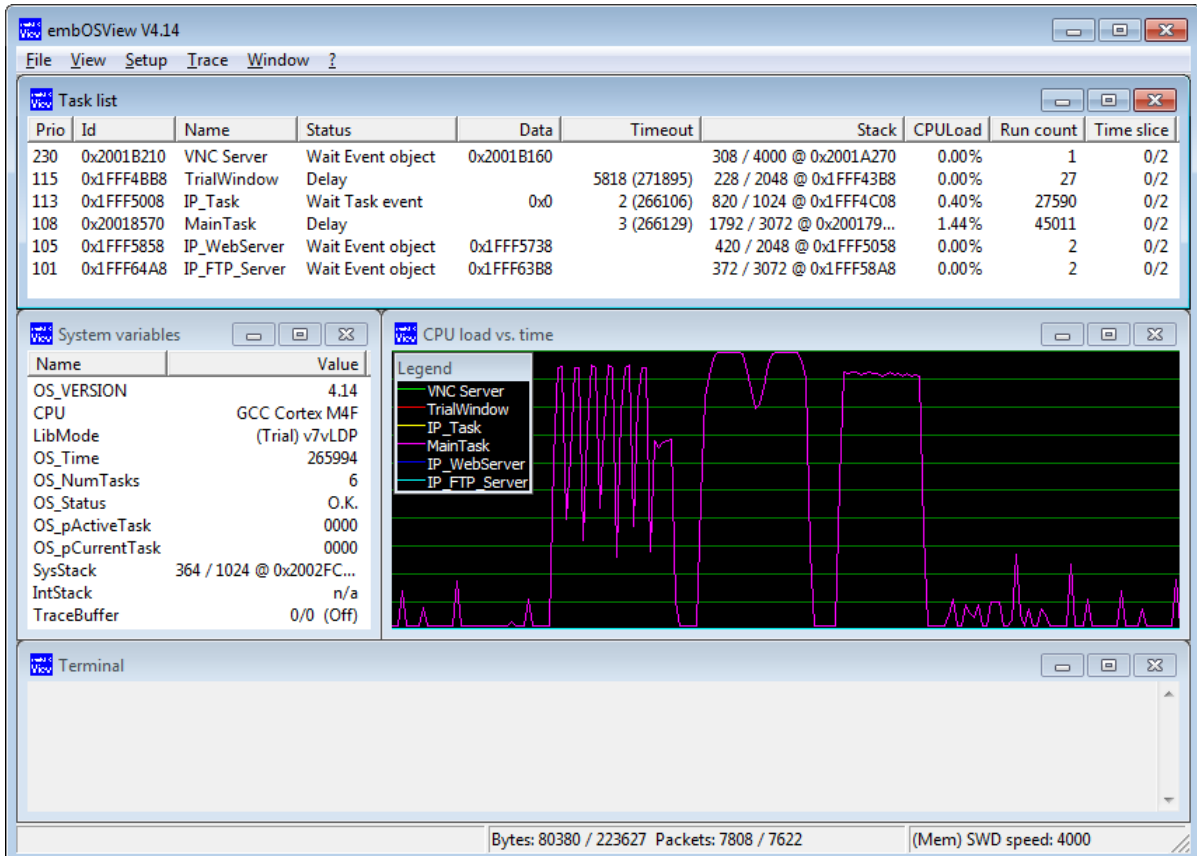
- In the lower left corner of the SEGGERDEMO slideshow, you will see the target's IP address.[2] Open a command prompt and try to ping your target. Your screen should look similar to the screenshot below.[3]


1. Only applies if UART is supported by the BSP
2. Depends on screen size
3. Only applies if the BSP contains embOS/IP

- The IP address of your board can also be identified via UDP. For this purpose the sample UDPDiscover, located at **Start\Windows\IP\UDPDiscoverGUI.exe**, can be used.[1]
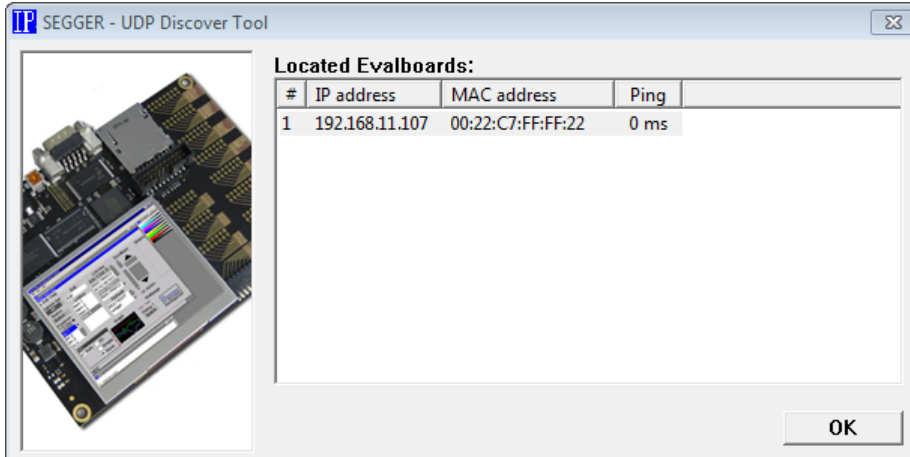
- Connect to the web server by entering **http://<target_ip>** into your web browser. Your screen should look similar to the screenshot below.[2]
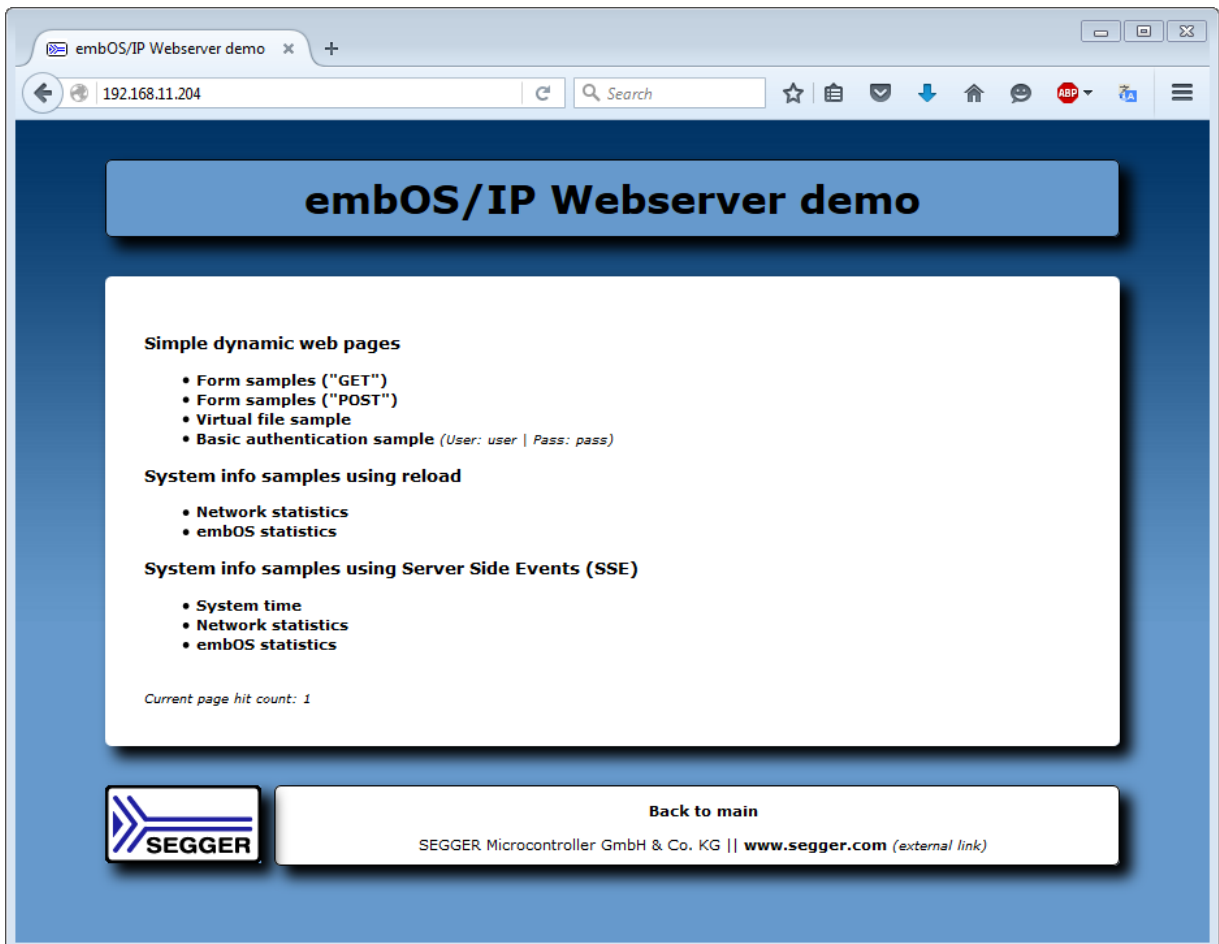
1. Only applies if the BSP contains embOS/IP
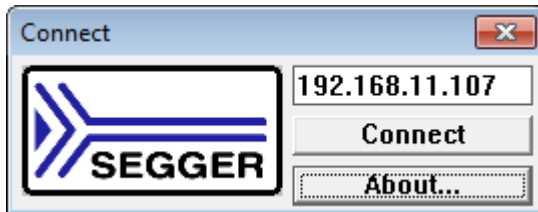2. Only applies if the BSP contains embOS/IP and a web server is included

- You can test the FTP server by connecting a FTP client to the target on port 21. You can do this via a command prompt following to the screenshot below.[1]

- A new mass storage device should be visible in your explorer.[2] If you want to change the memory configuration (e.g. RAMDisk, SD-Card, NAND-Flash…) include the corresponding config file (**Start\Setup\Exclude\FS_Config_*.c**) and exclude the current one.
- A VNC server is running in the background.[3] To connect to the server a VNC client, located at **Start\Windows\GUI\emVNC.exe**, can be used.

- After evaluating all parts of the SEGGERDEMO, you may want to evaluate the other samples located in the **Application** folder.

1.Only applies if the BSP contains embOS/IP and a FTP server is included
2.Only applies if the BSP contains emUSB Device and emFile
3.Only applies if the BSP contains embOS/IP and a VNC Server is included

2.3   Running a sample application

The folder **Application** includes a default application sample, which itself consists of either a single file or a subfolder, and a subfolder "Excluded", which contains all other application samples and is excluded from build. The default application sample may vary depending on the included middleware components: If emWin was included in the package, the SEGGERDEMO application sample is the default application. If emWin was not included, however, the default is either OS_StartLEDBlink or OS_Start2Tasks.

Each available sample contains a Task MainTask() and possibly other tasks. The MainTask() is always created from within the main() routine, which itself is located at Main.c in the folder **OS**. Main.c is required by all supplied sample applications.

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

/**********************************************************************
*
*       Prototypes
*
***********************************************************************
*/

#ifdef __cplusplus
extern "C" {     /* Make sure we have C-declarations in C++ programs */
#endif
void MainTask(void);
#ifdef __cplusplus
}
#endif

/**********************************************************************
*
*       Static data
*
***********************************************************************
*/

static OS_STACKPTR int Stack0[768];                   /* Task stack */
static OS_TASK         TCB0;                  /* Task-control-block */

/**********************************************************************
*
*       main()
*
* Function description
*   Application entry point
*/
int 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);                     /* Initially set LED             */
  /* You need to create at least one task before calling OS_Start() */
  OS_CREATETASK(&TCB0, "MainTask", MainTask, 100, Stack0);
  OS_Start();                        /* Start multitasking            */
  return 0;
}
```

To switch to another application sample than the default application, you may simply exclude that default application and include any other application instead.

Some embOS/IP and emUSB application samples are client/server applications and consist of one application for the embedded target and a second application for personal computers running Microsoft Windows. The latter are provided both in source code and as executable

within the **Windows** directory. To recompile the executable, an appropriate compiler is required: For example, suitable compilers are delivered with Microsoft Visual C++ 6.0 or Microsoft Visual Studio .Net.
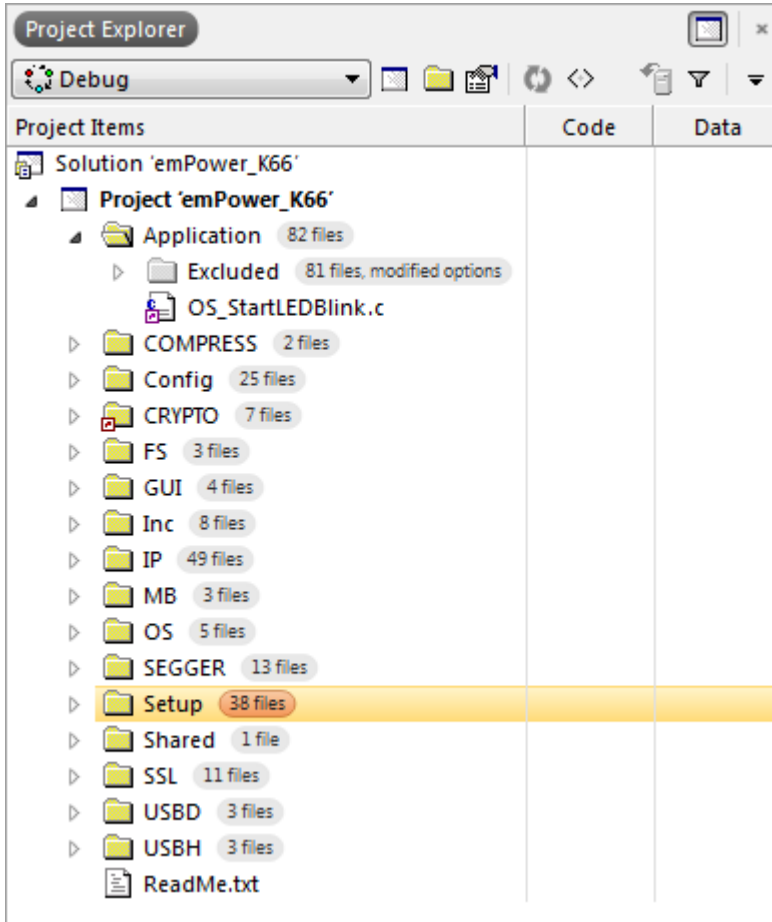
Chapter 3

Embedded Studio

This chapter contains additional information to start working with the sample applications and Embedded Studio.
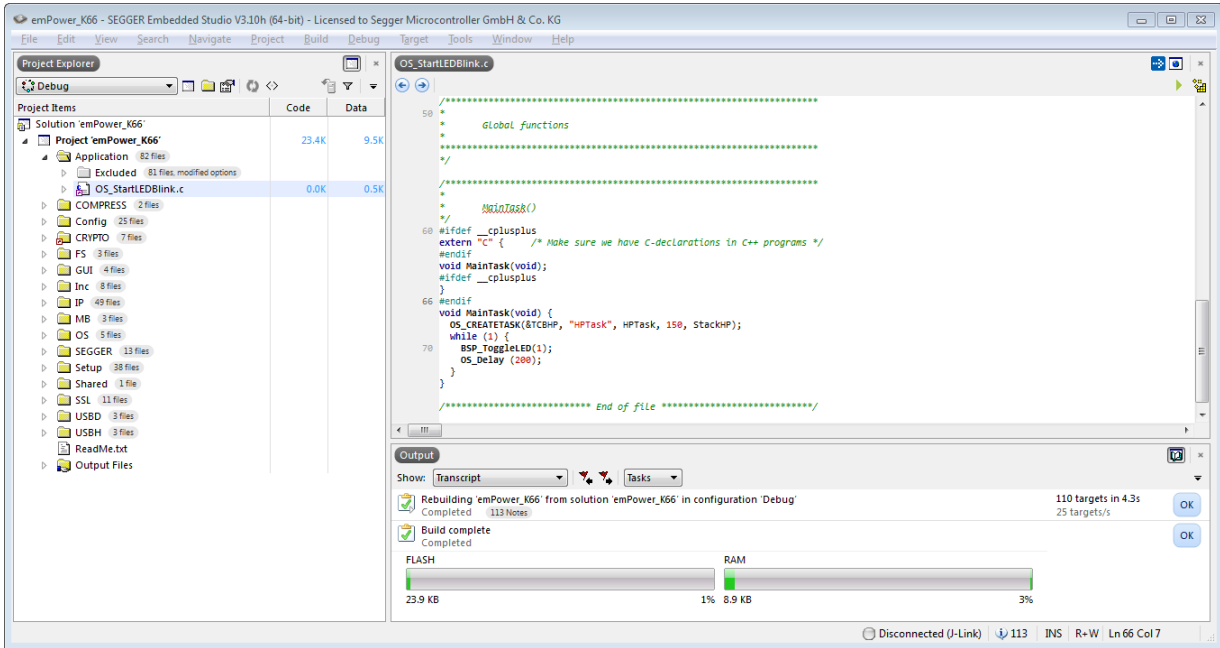
3.1   Changing the sample application

Open the **Application** folder and use "drag and drop" in the **Project Explorer** window to change the sample application. Move the included sample application (e.g. OS_StartLEDBlink.c) to the folder **Excluded** and move the favored sample application from the folder **Excluded** to the folder **Application**. This works also with a whole folder e.g. the SEGGERDEMO folder.
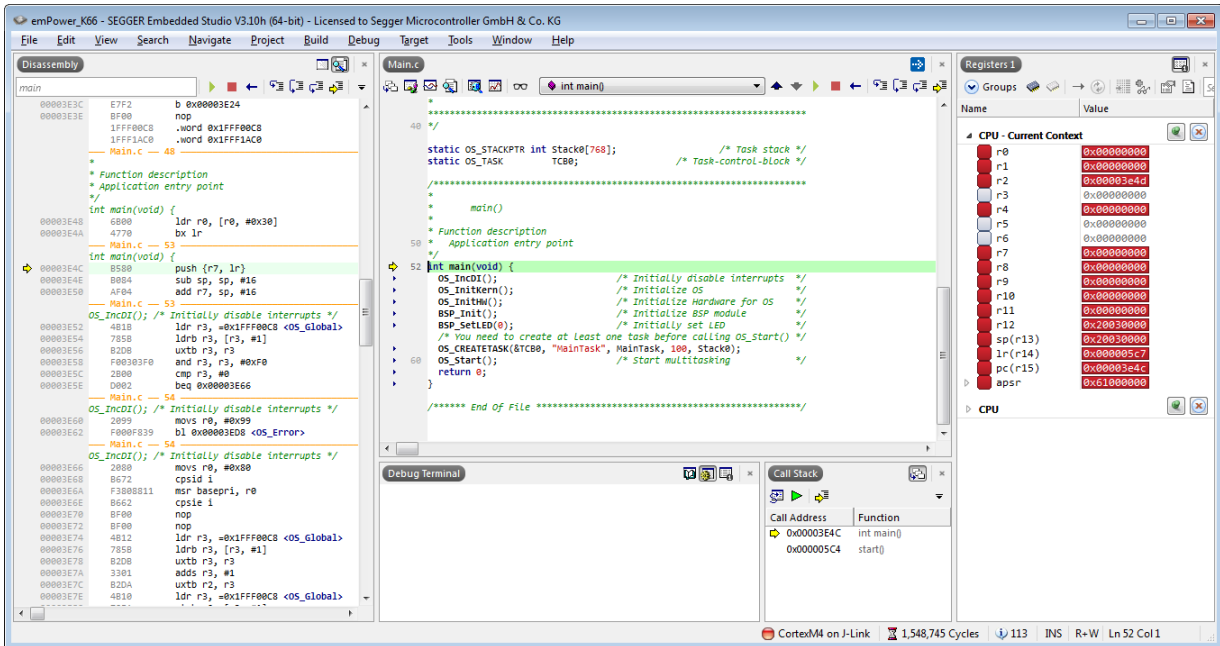
3.2　Build, download and run your application

Press [F7] to compile the selected application. The build log should report no errors or warnings.

You may download the application by pressing [F5]. As soon as your application halts at main(), press [F5] again to run the application.
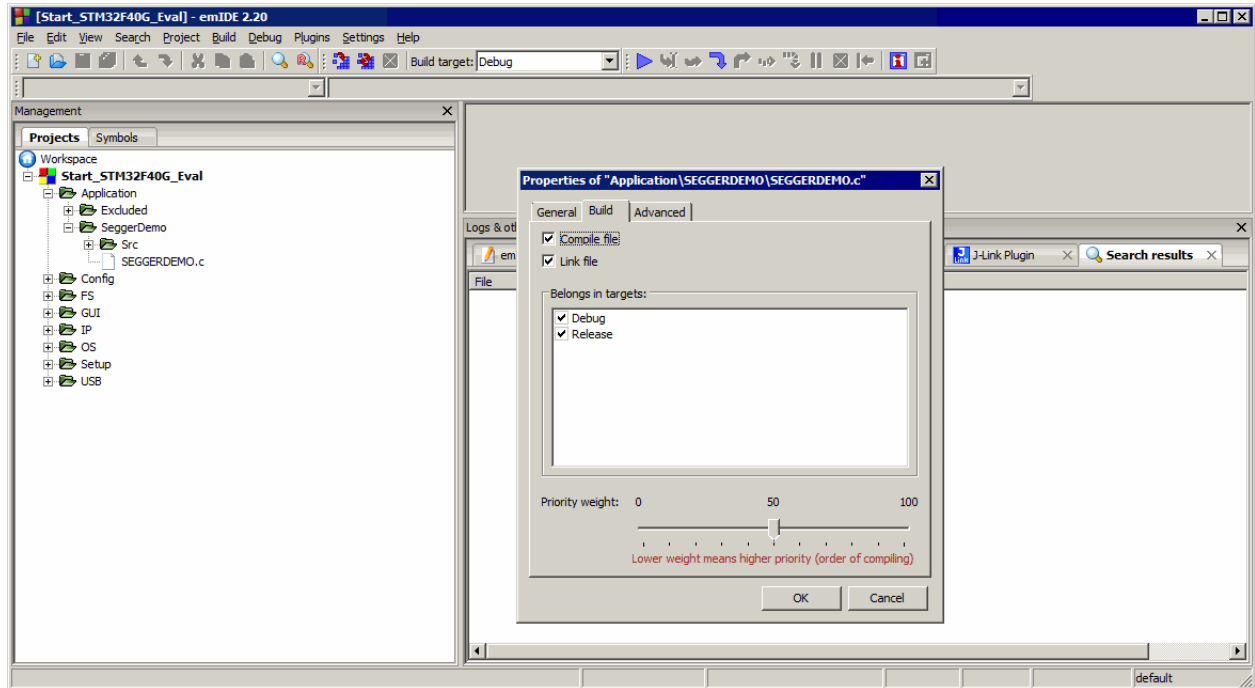
Chapter 4

emIDE

This chapter contains additional information to start working with the sample applications and emIDE.

4.1   Changing the sample application

To change the sample application, open the **Application** folder, open the context menu of the currently selected application with a right click and choose "Properties". Then select the tab "Build" in the properties window and un-check the upper check boxes, which are labeled "Compile file" and "Link file".

To include a sample application, go to **Application/Excluded** and repeat the previous steps on the file you wish to be included.

4.2   Build, download and run your application

Press [F7] to compile the selected application. The build log should report no errors or warnings.

You may download the application by pressing [F5]. When your application halts at main(), press [F5] once more to resume the application's execution.
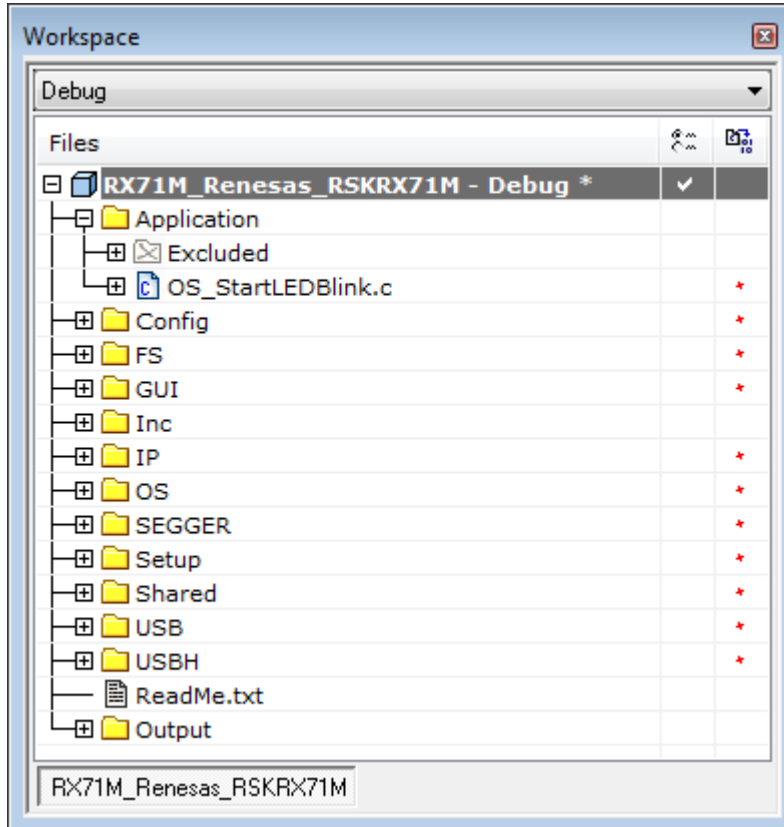
Chapter 5

IAR

This chapter contains additional information to start working with the sample applications and IAR Embedded Workbench.

5.1   Changing the 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_StartLEDBlink.c) to the folder **Excluded** and move the favored sam-
ple application from the folder **Excluded** to the folder **Application**. This works also with
a whole folder e.g. the SEGGERDEMO folder.

Changing an application is only possible if no debug session is running.

5.2   Build, download and run your application

Press [F7] to compile the included application. The build log should report no errors and
no warnings.
After a successful build you can download the application by pressing [ctrl + D].
When the application halts at main() press [F5] to run your application.

Chapter 6

CrossWorks

This chapter contains additional information to start working with the sample applications and Rowley's CrossWorks.

6.1   Changing the sample application

To change the sample application open the **Application** folder, open the context menu of the current application (e.g. OS_StartLEDBlink.c) with a right click and choose "Exclude from build" to exclude it.

To include a sample application go to **Application/Excluded** and repeat the previous steps on the file you wish to be included.

To exclude a whole folder (e.g. the SEGGERDEMO) right click on it and open the properties window. Type into the properties search field (located at the top of the properties window) "exclude" and set the Build option "Exclude from build" to "yes".
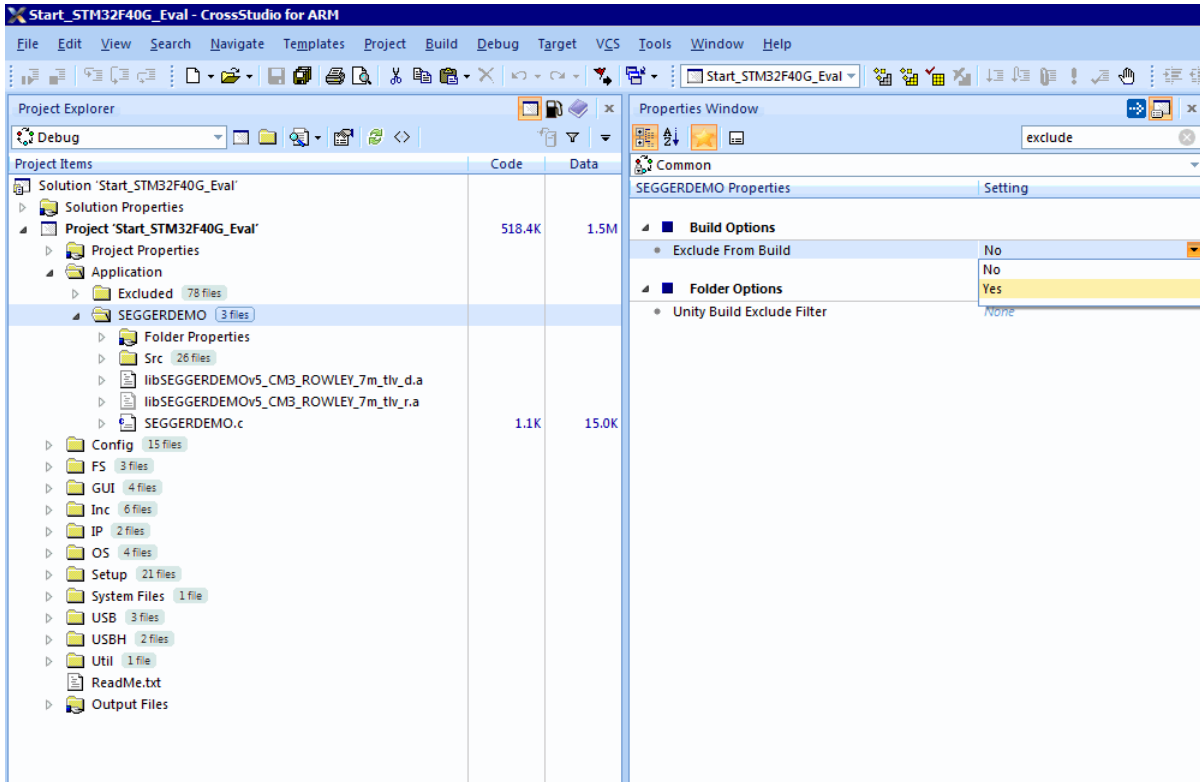
6.2   Build, download and run your application

Press [F7] to compile the included application. The build log should report no errors and no warnings.

Before you can download the application make sure you are connected to the target. To connect to a target double click on an entry in the "Targets" window or use the appropriate entry in the context menu.

The targets window of CrossWorks should look similar to the picture below.
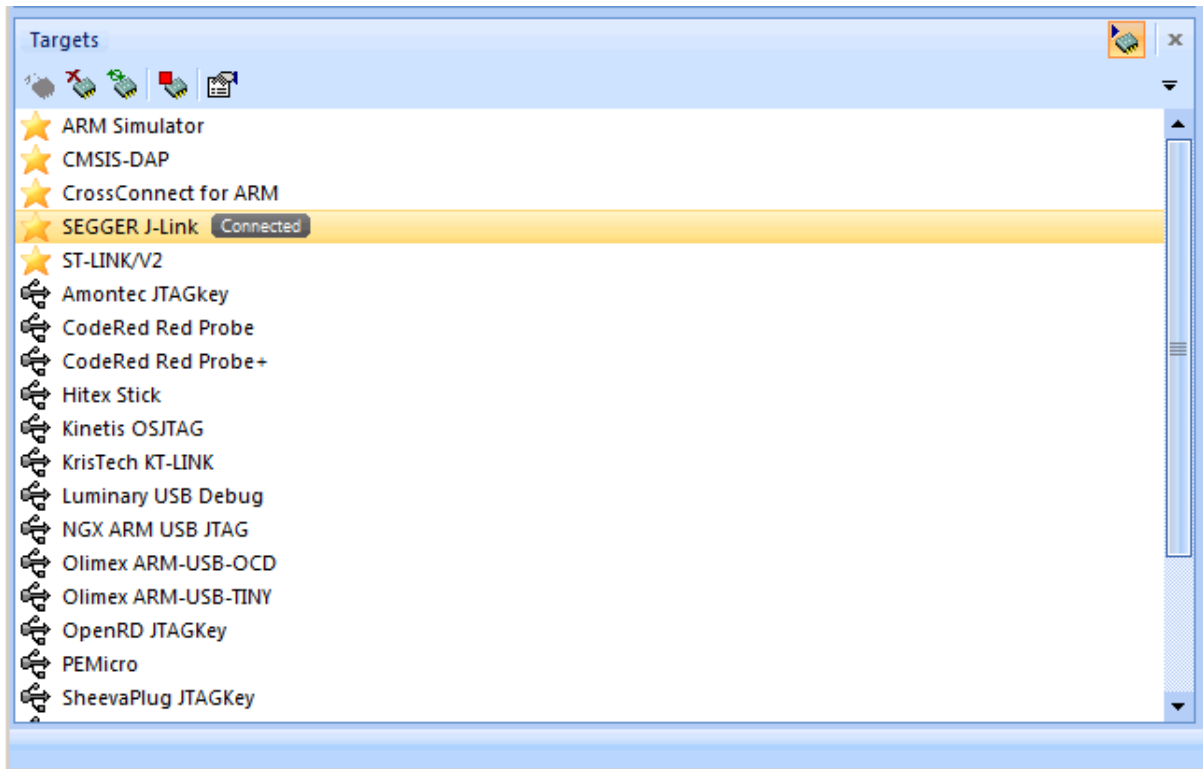
If you are connected to your target you can download the application by pressing [F5]. As soon as your application halts at main() press [F5] again to run the application.
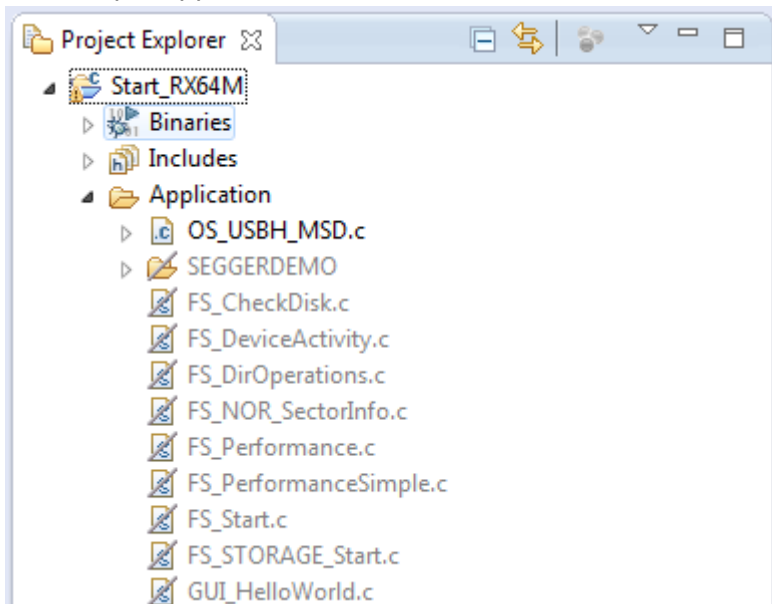
Chapter 7

Renesas e2 studio

This chapter contains additional information to start working with the sample applications and Renesas e^2 studio.

7.1   Changing the sample application

Open the **Application** folder in the **Project Explorer** window of the e^2 studio to change the sample application.

Right-click the already included application and select **Properties**, in the properties menu check the **Exclude from build** checkbox.
Right-click the application you wish to evaluate, select **Exclude from build...**, deselect all available configurations and click **OK**.

This works also with a whole folder e.g. the `SEGGERDEMO` folder.
Changing an application is only possible if no debug session is running.

7.2 Build, download and run your application

Press [ctrl + b] to compile the included application. The build log should report no errors or warnings.

After a successful build you can download the application by opening the **Debug Configurations** window.
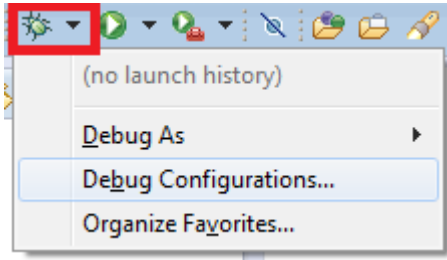
In the **Debug Configurations** window select the appropriate debug configuration and double-click it:
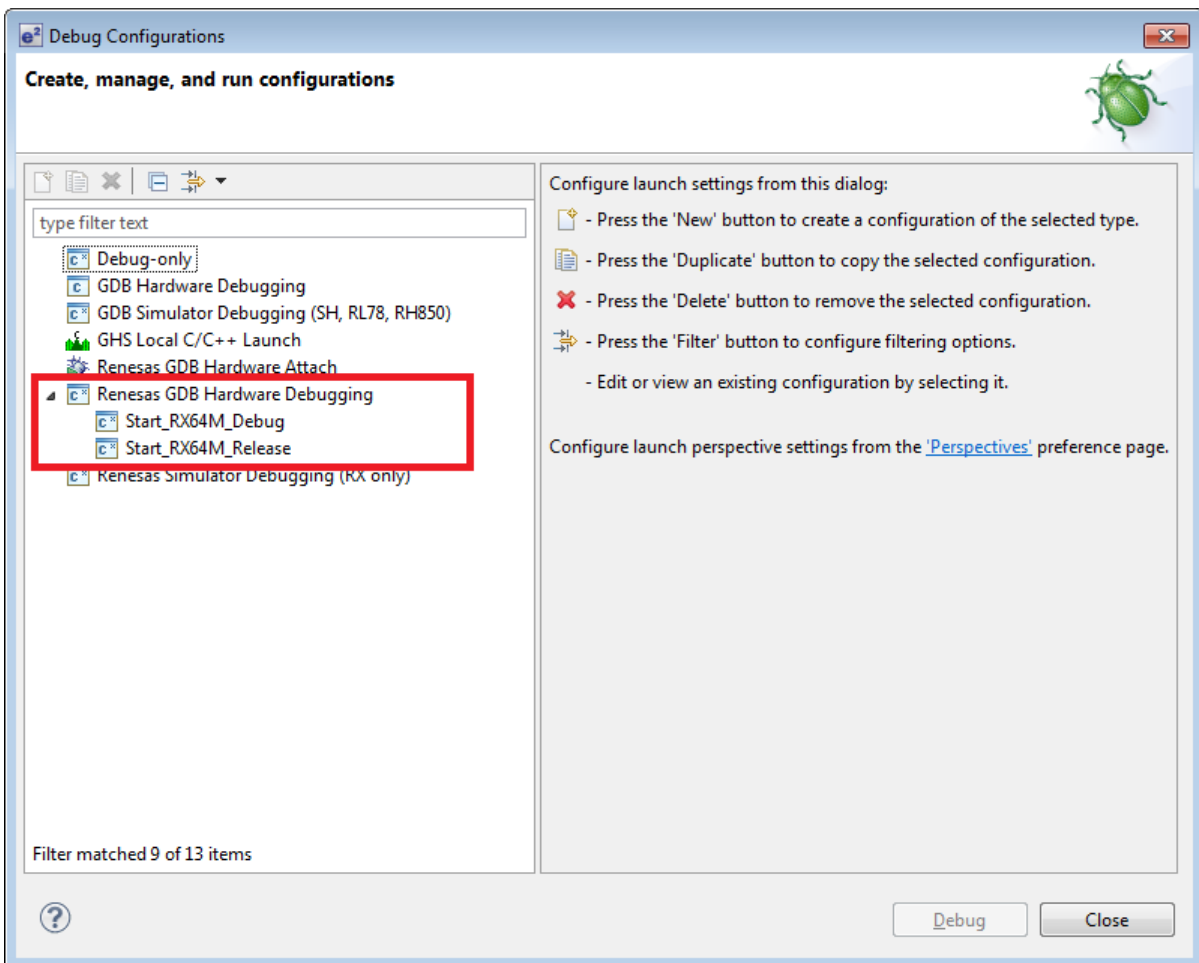
As soon as your application halts at main() press [F8] to run the application.
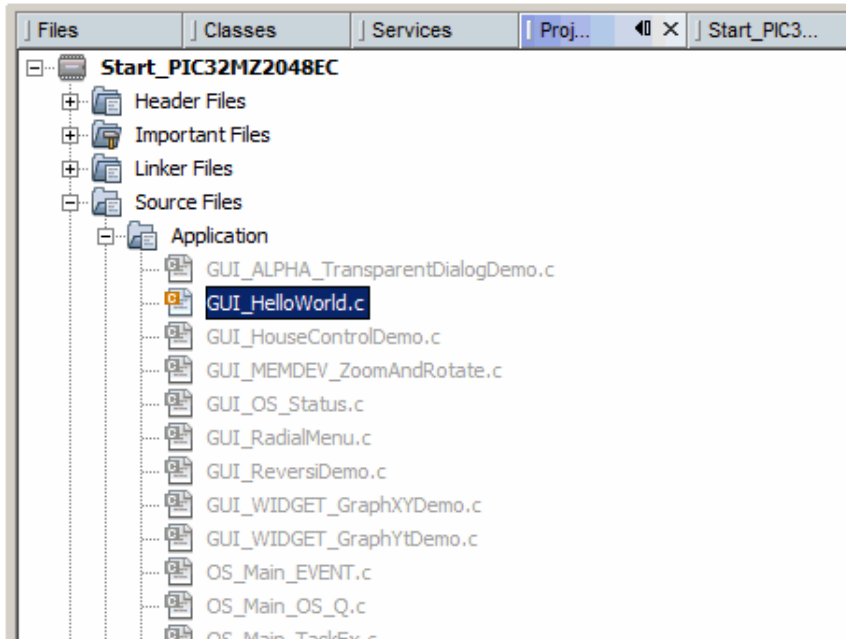
Chapter 8

Microchip MPLAB X

This chapter contains additional information to start working with the sample applications and Microchip MPLAB X.

8.1 Changing the sample application

Open the **Application** folder in the **Project Explorer** window of MPLAB X to change the sample application.

Right-click the already included application and select **Exclude from build...**, select all available configurations and click **OK**.
Right-click the application you wish to evaluate, select **Exclude from build...**, deselect all available configurations and click **OK**.

This works also with a whole folder e.g. the `SEGGERDEMO` folder.
Changing an application is only possible if no debug session is running.

8.2   Build, download and run your application

To download and run the included application press [F6]. Then the project will get compiled and downloaded. The build log should report no errors or warnings.

To run a debug session open the **Debug** menu and choose **Debug Main Project**. Now the project gets build with debug information and downloaded to the target.
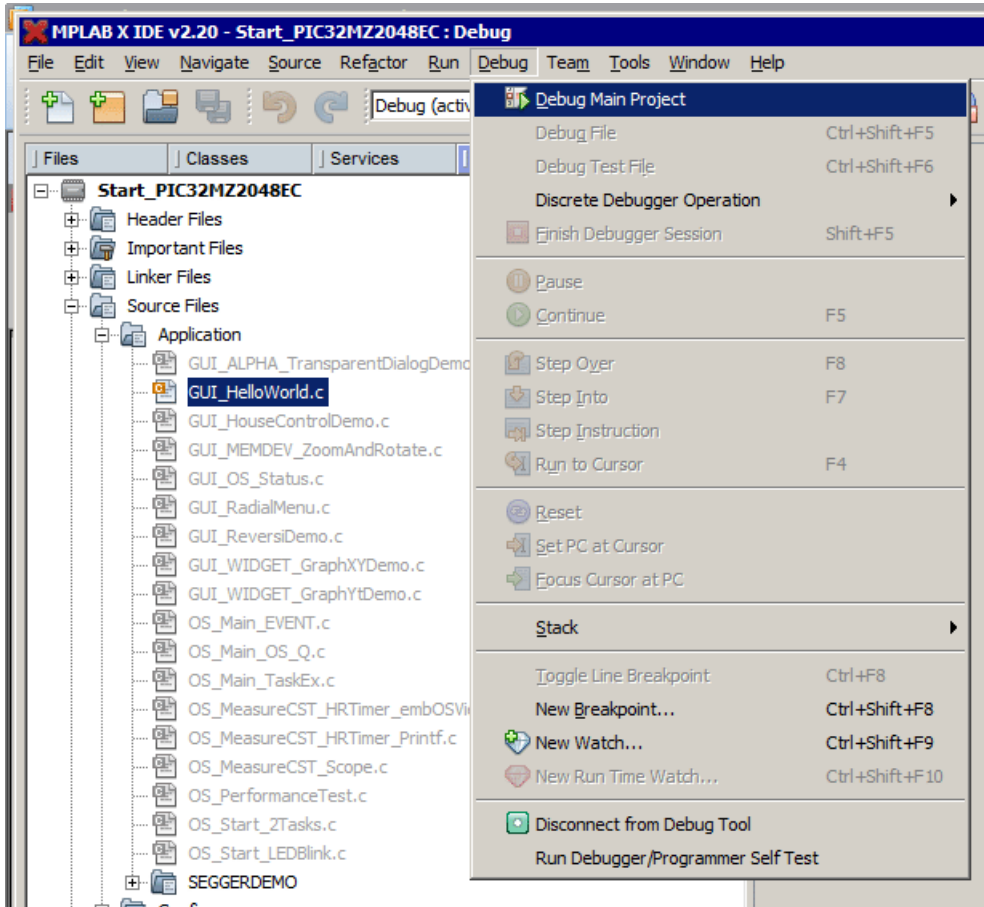
To choose another **Debug Configuration** click in the configuration drop down menu and choose a **Debug Configuration**:
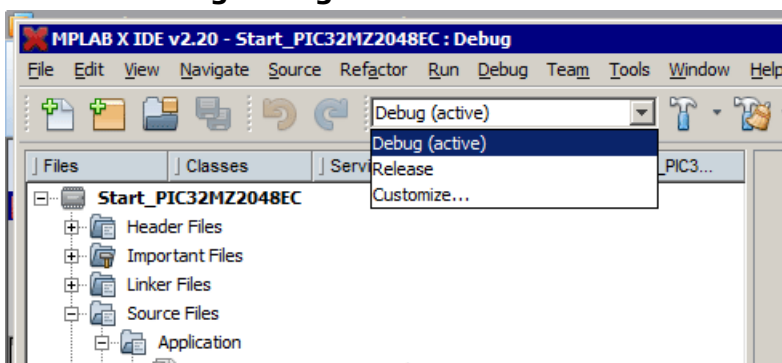
These are some useful shortcut keys:
- [Ctrl + F8] - Toggle line breakpoint
- [F5] - Run the application
- [F7] - Step into
- [F8] - Step over

Chapter 9

Keil MDK

This chapter contains additional information to start working with the sample applications and Keil MDK.
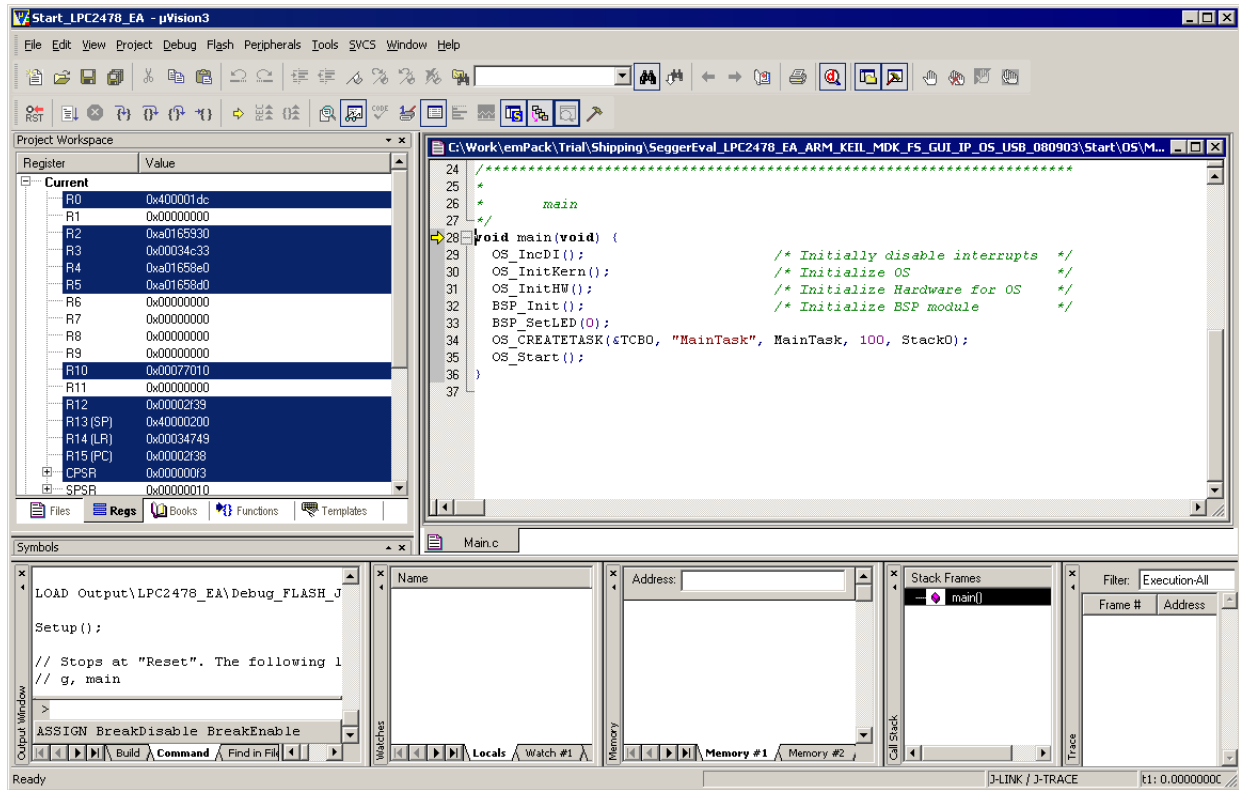
9.1   Changing the sample application

To change the sample application, open the **Application** folder and use "drag and drop" in the **Project** window of the KEIL MDK IDE. Move the included sample application (e.g. SEGGERDEMO.c) in the **Application\Excluded** folder and move the favored sample application from the **Application\Excluded** folder to the **Application** folder.
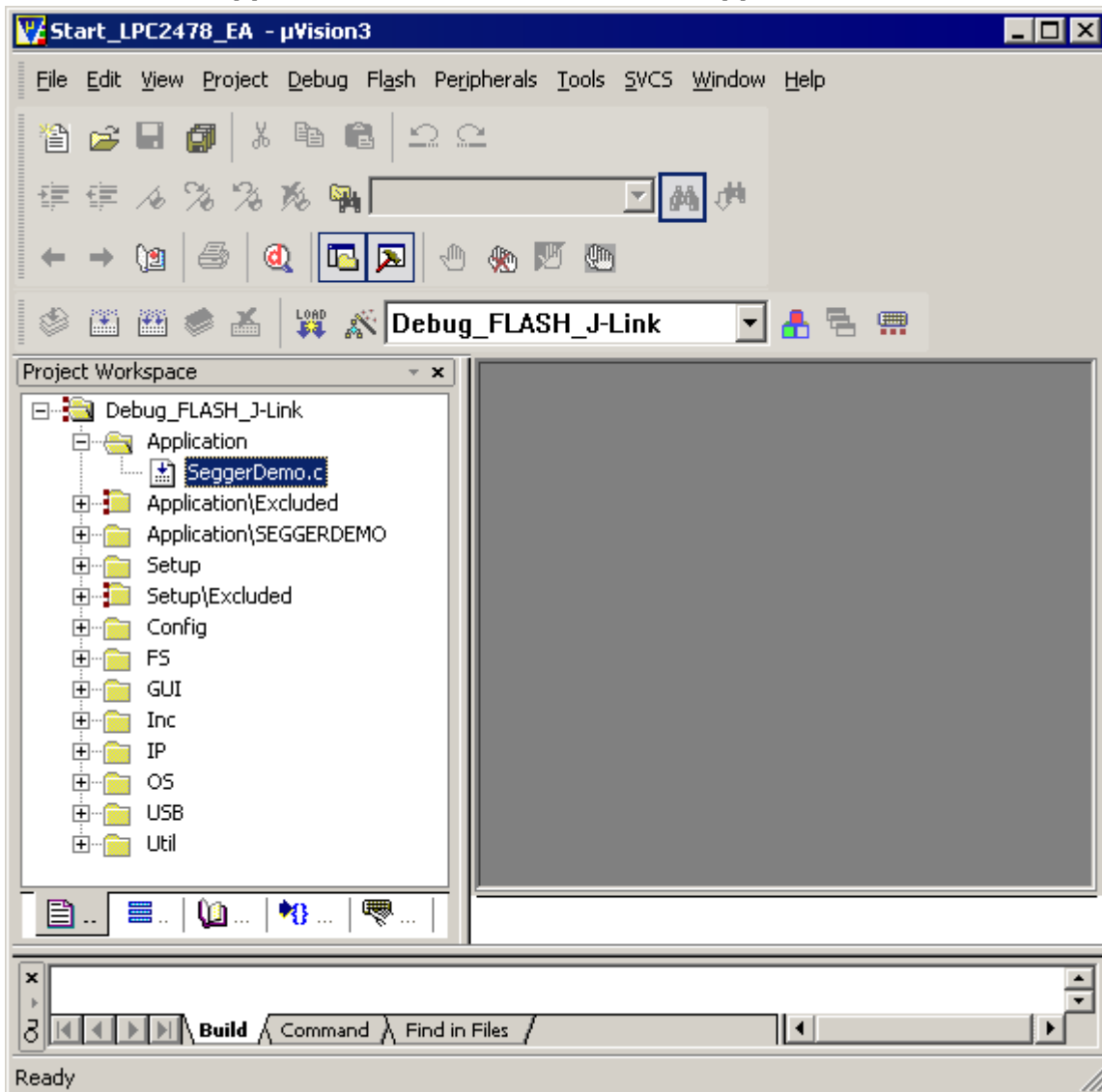
9.2   Build, download and run your application

Press [F7] to compile the selected sample application. The build log should report no errors or warnings.
Press CTRL+[F5] to download the application and start a debug session. After download your screen should be similar to the screenshot below, halting at `main()`.

As soon as your application halts at main(), press [F5] again to run the application.

Chapter 10

Prebuild sample application

This chapter explains how to download the prebuild application samples into your target using the `SEGGER J-Link` and the `J-Link Commander`.

10.1   Using J-Link Commander

If you are using a SEGGER J-Link as debug probe, it is possible to download the prebuild sample application via J-Link Commander by following the steps described in this chapter. Connect your target to your host PC via J-Link, ensure your board is powered and start the batch file **StartBinary.bat**. The download of the application will now be executed.

After the download has been finished, the prebuild application will start automatically.

For additional information on how to use J-Link or J-Link Commander, please refer to the J-Link documentation (UM08001). The J-Link documentation (UM08001) and J-Link Commander are part of the J-Link Software and Documentation Package, which may be downloaded from *segger.com/jlink-software.html* .

Chapter 11

Literature and references

This chapter lists documents which may be useful to gain a deeper understanding of technical details.

Reference	Product	Description
UM01001	embOS	This document gives information about using the generic parts of embOS.
UM010xx	embOS	This document gives information about using the CPU and compiler specific parts of embOS. It is included in respective embOS trial packages available from SEGGER.
UM02001	emFile	This document gives information about using SEGGER emFile.
UM03001	emWin	This document gives information about using SEGGER emWin.
UM07001	embOS/IP	This document gives information about using SEGGER embOS/IP.
UM08001	J-Link / J-Trace	This document gives information about using SEGGER J-Link / J-Trace.
UM09001	emUSB Device	This document gives information about using SEGGER USB Device.
UM10001	emUSB Host	This document gives information about using SEGGER USB Host.
UM14001	emModbus	This document gives information about using SEGGER Modbus.
UM15001	emSSL	This document gives information about using SEGGER emSSL.

If not stated otherwise, all manuals are publicly available for download (see links above) and are also shipped within the package, in the **Doc** folder.
The emSSL manual is available upon request.

Chapter 12

Appendix

12.1   Software components in the package

12.1.1   emFile

emFile is SEGGER's embedded file system that can be used on any storage medium for which you can provide basic hardware access functions.
emFile is a high-performance library that has been optimized for minimum memory consumption in RAM and ROM, high speed and versatility.

The emFile documentation may be found at **Doc\UM02001_emFile.pdf**.

12.1.2   emModbus

emModbus is SEGGER's implementation of the Modbus protocol, supports communication via UART (ASCII, RTU) and Ethernet (Modbus/TCP and Modbus/UDP) and is capable to communicate with any Modbus compliant device.

The emModbus documentation may be found at **Doc\UM14001_emModbus.pdf**.

12.1.3   emWin

emWin is SEGGER's embedded Graphical User Interface (GUI) using a feature rich API and providing an efficient, processor- and LCD-controller-independent GUI for any application that operates with a graphical LCD. It may be adapted to any size physical and virtual display with any LCD controller and CPU.

The emWin documentation may be found at **Doc\UM03001_emWin.pdf**.

12.1.4   embOS

embOS is SEGGER's embedded priority-controlled multitasking system. It is designed to be used as an embedded operating system for the development of real-time applications and has been optimized for minimum memory consumption in both RAM and ROM, as well as high speed and versatility.

The embOS documentation may be found at **Doc\UM01001_embOS.pdf** and **Doc\UM010xx_embOS_<cpu>_<compiler>.pdf**.

12.1.5   embOS/IP

embOS/IP is SEGGER's embedded TCP/IP stack. It is a CPU independent, high-performance TCP/IP stack that has been optimized for speed, versatility and small footprint.

The embOS/IP documentation may be found at **Doc\UM07001_embOSIP.pdf**.

12.1.6   emUSB Device

emUSB Device is SEGGER's embedded USB Device stack. It features bulk communication as well as device classes such as MSD, CDC or HID.

The emUSB Device documentation may be found at **Doc\UM09001_emUSB.pdf**.

12.1.7   emUSB Host

emUSB Host is SEGGER's embedded USB Host stack. It features bulk communication as well as device classes such as MSD or HID, and also supports external hubs.

The emUSB Host documentation may be found at **Doc\UM10001_emUSBH.pdf**.

12.1.8   emSSL

emSSL is a SEGGER software library that enables secure connections across the Internet. emSSL offers both client and server capability.

The emSSL documentation may be found at **Doc\UM15001_emSSL.pdf**.

> **Note**
>
> The emSSL software is classified as dual-use good according to Category 5, Part 2 "Information Security" of EC Regulation No 428/2009. Any export or transfer of the software with a destination outside the European Union requires an export permission (Art. 22 Abs. 8 und 10 EG-Dual-Use-VO).

Therefore, two versions of each SEGGER Eval Software package are available:
- The standard version available for download via *SEGGER website: Evalboards*
- The complete version available up on request via info@segger.com

The standard version does not contain export restricted software components.

12.1.9   emCompress

emCompress is a compression system that is able to reduce the storage requirements of data that must be embedded into an application.

The emCompress documentation may be found at **Doc\UM17001_emCompress.pdf**.

12.2   Package structure

Each component of SEGGER Eval Software is provided in library form with additional header files; sample applications are provided as source code.

In order to get started, it is neither necessary nor recommended to apply changes to the file structure of the SEGGER Eval Software package. Any desirable changes, e.g. changing the current sample application, should be done from inside the IDE.

The root directory of the package includes the folders listed below.

Directory	Contents
Application	Various application samples for the middleware components
BSP	Ready-to-use example project(s) using the middleware libraries as well as the Board Support Package (BSP). The BSP consists of all files that are required to initialize the target hardware, or to build an executable for the target hardware.
COMPRESS	emCompress header files and libraries
CRYPTO	emCrypt header files and libraries
Doc	Documentation of the SEGGER Eval Software and of the included middleware components
FS	emFile header files and libraries
GUI	emWin header files and libraries
IP	embOS/IP header files and libraries
MB	emModbus header files and libraries
OS	embOS header files and libraries
Prebuild	Prebuild binary of the SEGGERDEMO
SECURE	emSecure header files and libraries
SEGGER	Utility functions that are not directly related to one of the included middleware components
SSH	emSSH header files and libraries
SSL	emSSL header files and libraries
USB-D	emUSB Device header files and libraries
USB-H	emUSB Host header files and libraries
Windows	Example applications for Microsoft Windows and the drivers required for USB functionality. Refer to the User and reference guide for emUSB [UM09001] for more information about the installation of these drivers.

Furthermore, the `License.txt` file and this application note are located at the root directory.

The middleware (like FS, GUI, IP….) folders all include the following subfolder structure:

Directory	Contents
Conf	Configuration files
Inc	Header files
Lib	Library files
OS	OS abstraction layers