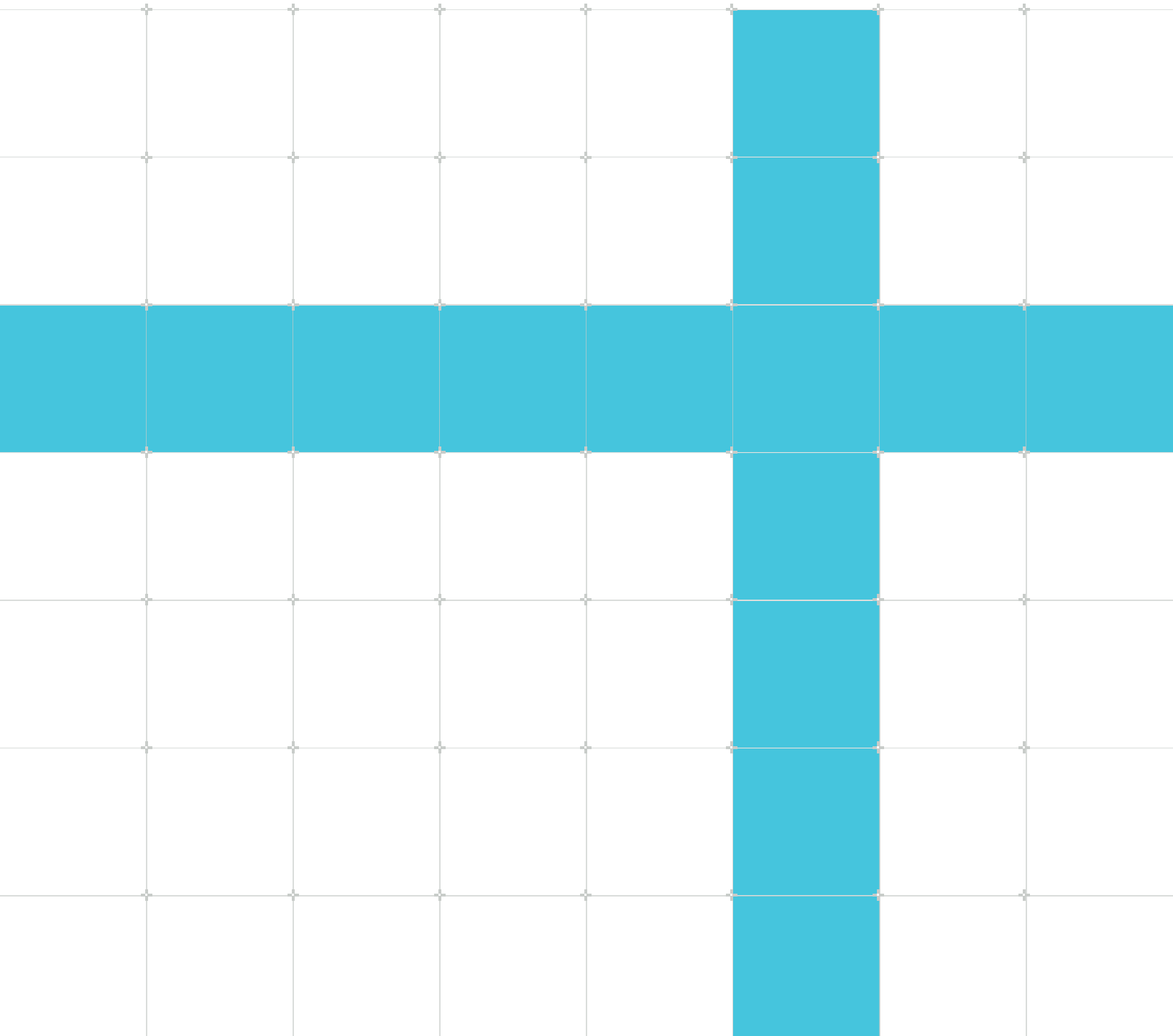
arm

Hello World in C for Bare-Metal Targets

Version 1.0

Issue 01
102647_0100_01_en



Hello World in C for Bare-Metal Targets

Release information

Document history

Issue	Date	Confidentiality	Change
0100-01	16 September 2022	Non-Confidential	Initial release

Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly

or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Feedback

Arm® welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

1. Introduction

This tutorial takes you through creating, configuring, and building a simple bare-metal program using Arm DS-5. To run your application once it is built, the tutorial then takes you through the steps of configuring a debug connection to a system model implemented in software.

After installing and acquiring a license to work with DS-5, this tutorial takes you through creating, configuring, and building a simple bare-metal program.

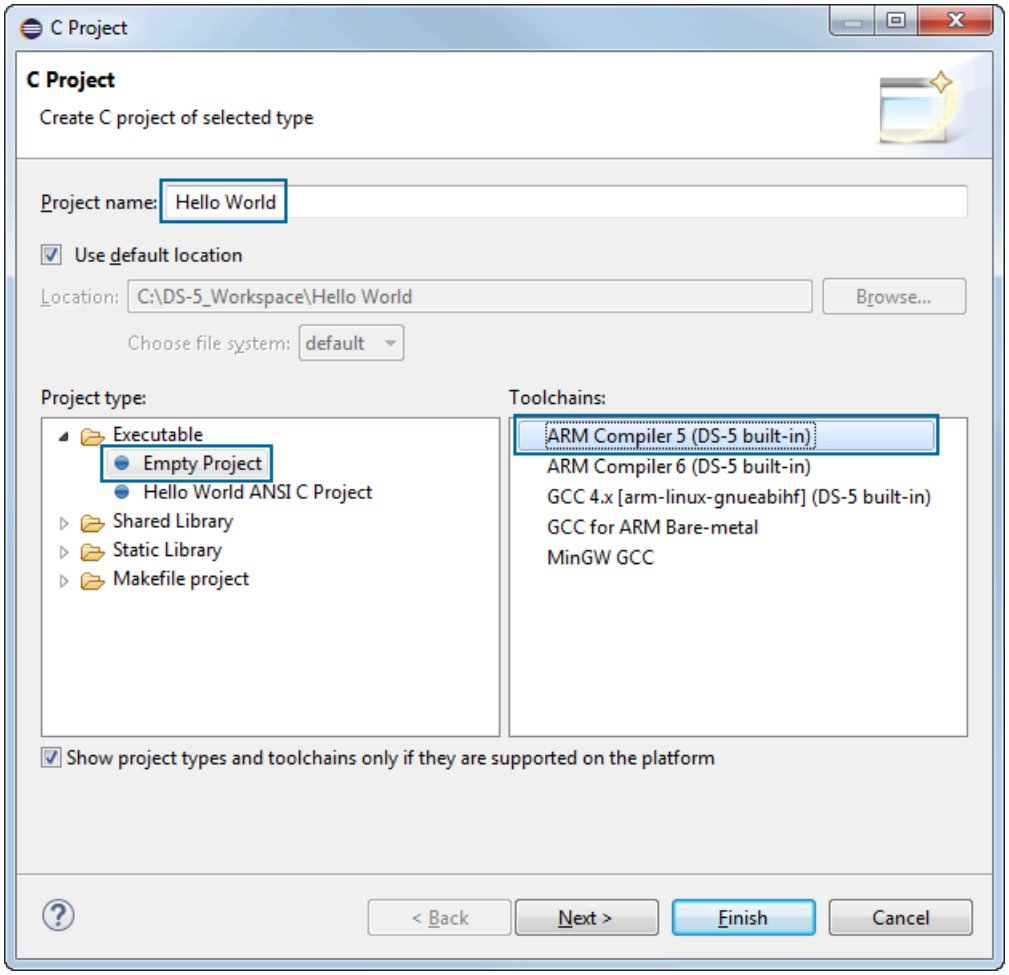
To run your application once it is built, the tutorial then takes you through the steps of configuring a debug connection to a system model implemented in software. These models are called Fixed Virtual Platforms (FVP) and some are provided with DS-5. This tutorial uses the `VE_Cortex_A9x1`FVP model which is based on the Cortex-A9 processor.

2. Creating a New C Project

The following steps help you create a new C Project.

1. From the DS-5 main menu, select File > New > C Project to display the C Project dialog.
2. In the C Project dialog:
   a. In Project name field, enter HelloWorld as the name of your project.
   b. Under Project type, select Executable > Empty Project. When selecting the Executable option, the toolchain assumes that the application is executed directly on the hardware instead of on top of a complex operating system such as Linux.
   c. Under Toolchains, select Arm Compiler 5.

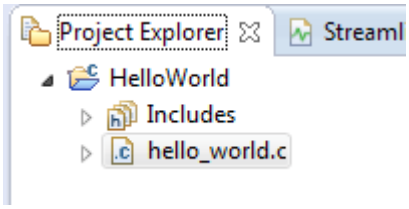
Figure 2-1: C project dialog options set v5211



Learn more about the Arm Compiler toolchain.

   d. Click Finish to create a C project called Hello World. You can view the project in the Project Explorer view.

Figure 2-2: HelloWorld File In Project Explorer



3. Specifying a RAM Base Address

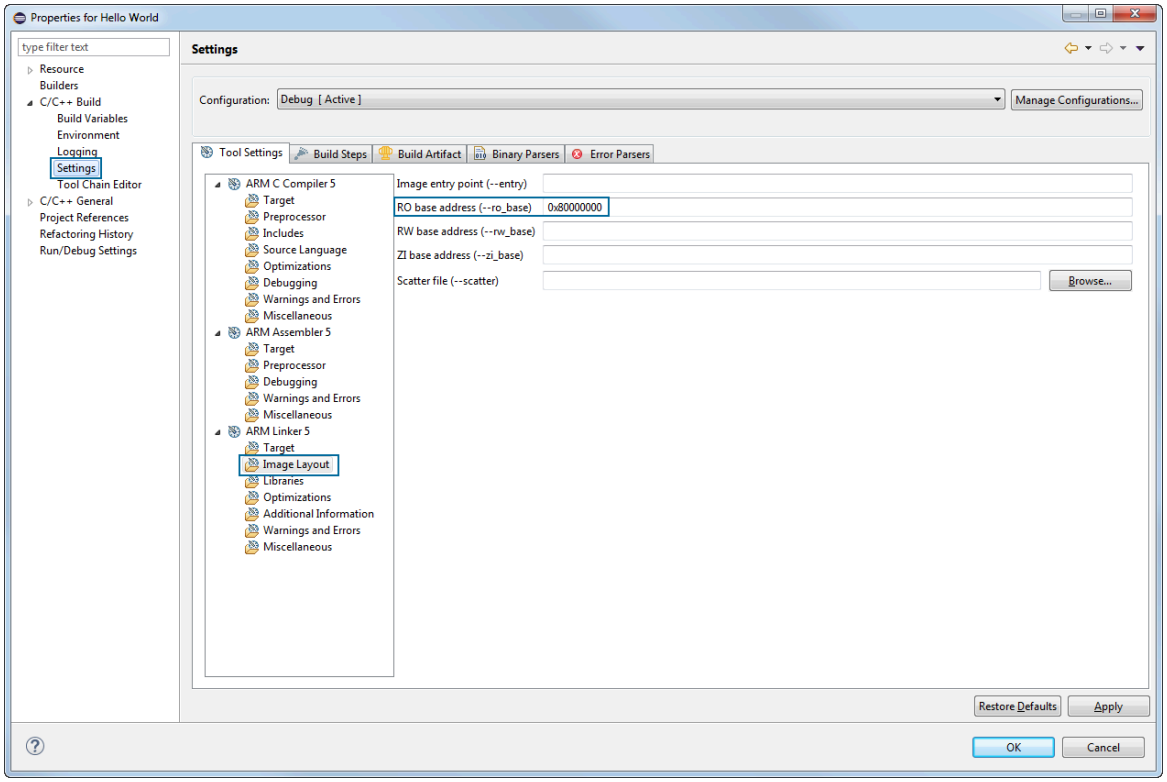
To load and execute the application on the target, before compiling the application, we need to tell the linker the target RAM base address. This ensures that the application is built correctly for the particular target.

The VE global model memory map contains the memory address details required for the VE FVP model used in this tutorial.

We can see that the memory address range for VE FVP models (4GB DRAM (in 32-bit address space)) is between `80000000` and `FFFFFFFF`. This gives us the RAM base address as `0x80000000`.

1. In Project Explorer, right-click the project and select Properties.
2. In the Properties dialog:
   a. Browse to C/C++ Build > Settings.
   b. Under the Tool Settings tab, browse to Arm Linker 5 > Image Layout.
   c. In the RO base address (–ro_base) field, enter `0x80000000`.

Figure 3-1: RO base address `0x80000000`



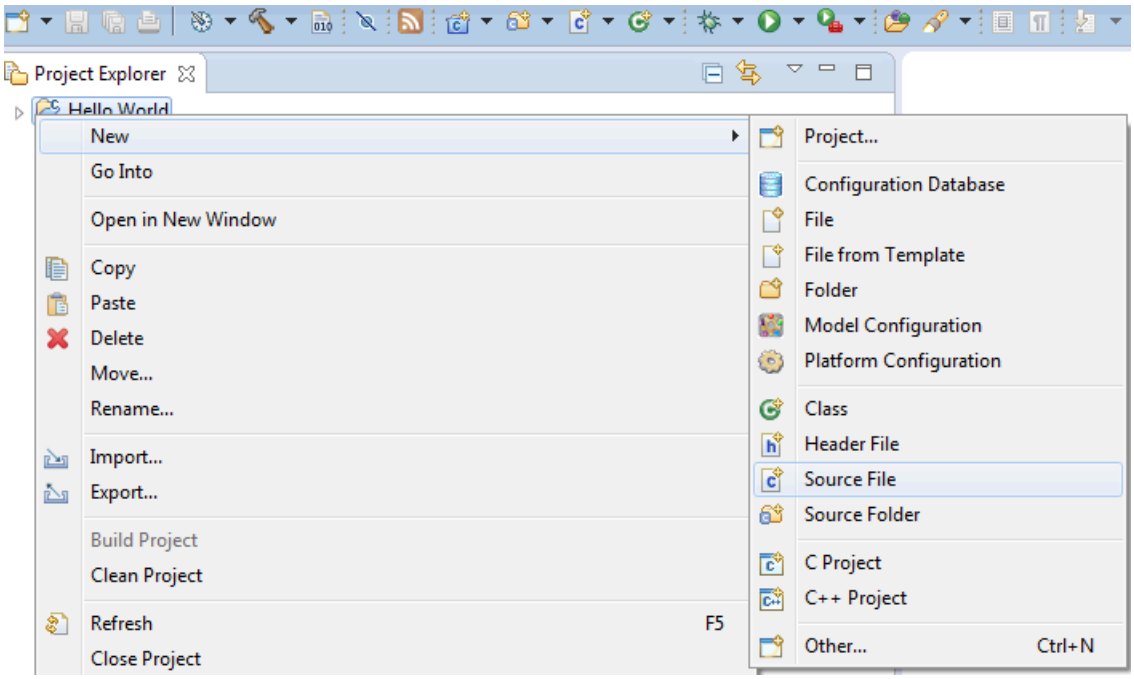
   d. Click OK to close the dialog and apply the changes.

4. Creating the Source Code and Building the Project

The following steps helps you create the source code and build the project:

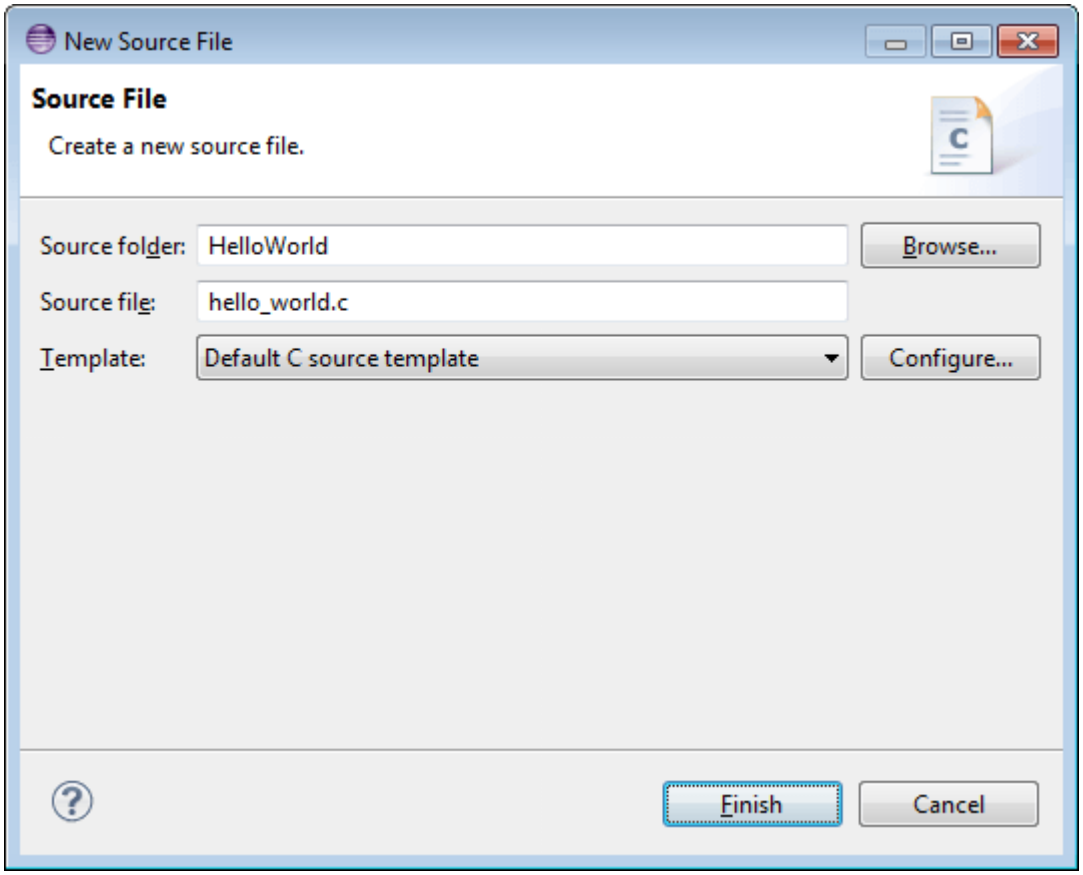
1. In the Project Explorer view, right-click the Hello World project and select New > Source File.

Figure 4-1: New Source File



2. In the New Source File dialog, enter the file name `hello_world.c`.

Figure 4-2: HelloWorld Enter FileName

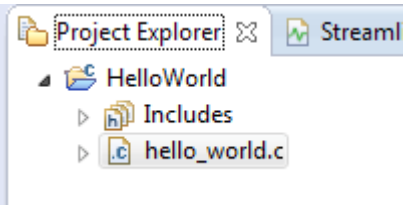


3. Click Finish to create the source file and open it in the code editing view.

Figure 4-3: Code Editing View

The source file is also visible in the Project Explorer view, under the Hello World project.

Figure 4-4: Hello World File In Project Explorer



4. Add the following code to the new source file, and press CTRL+S to save it.

```
#include <stdio.>

 int main() {
   printf("Hello World\n");
 }
```

1. In the Project Explorer view, right-click on the Hello World project and select Build Project.

You can view the output image `hello_world.axf` in the Debug folder under the HelloWorldproject.

The `.axf` file contains both the object code and debug symbols that enable the debugger to perform source-level debugging.