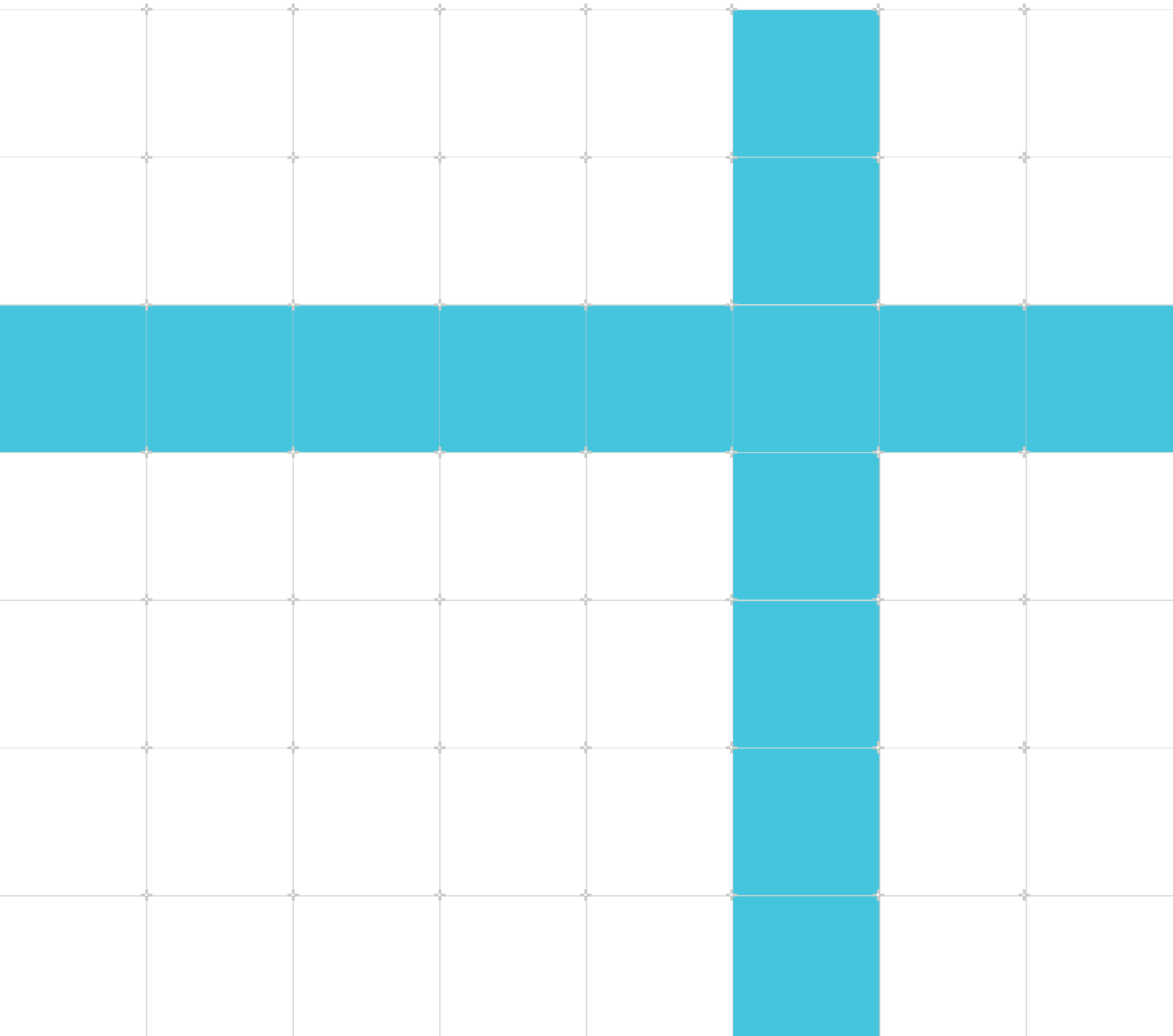
Debugging Hello World in C on a Fixed Virtual Platform

Version 1.0

Issue 00
102628_0100_00_en



Debugging Hello World in C on a Fixed Virtual Platform

Release information

Document history

Issue	Date	Confidentiality	Change
0100-00	1 January 2020	Non-Confidential	Initial release

Proprietary Notice

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

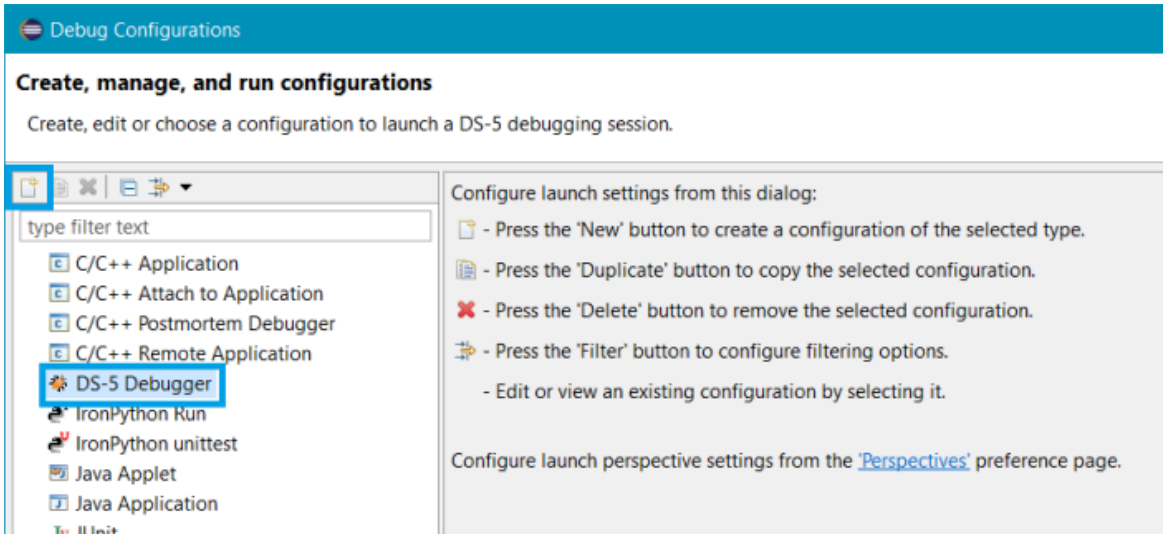
Once you have created the project and built the code, launch the debugger to run the application on one of the Fixed Virtual Platforms (FVP) provided with DS-5.

For this tutorial, we use a Cortex-A9 Fixed Virtual Platform (FVP) which is provided with DS-5.

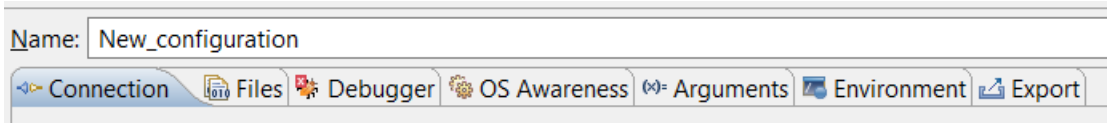
2. Create a DS-5 debug configuration and connect to an FVP

The following steps help you create a DS-5 debug configuration and connecting to an FVP.

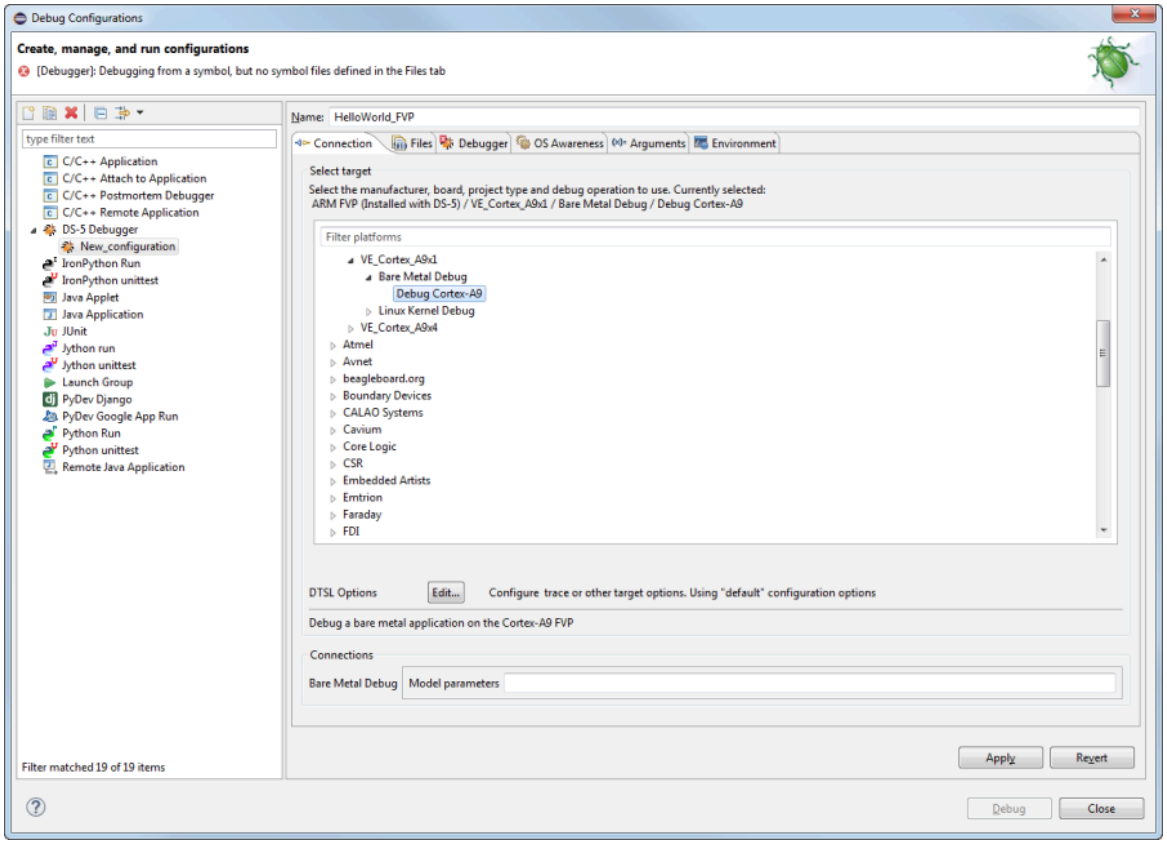
1. From the DS-5 main menu, select Run > Debug Configurations.
2. In the Debug Configurations dialog:
   a. Select DS-5 Debugger.
   b. Click the New launch configuration button.



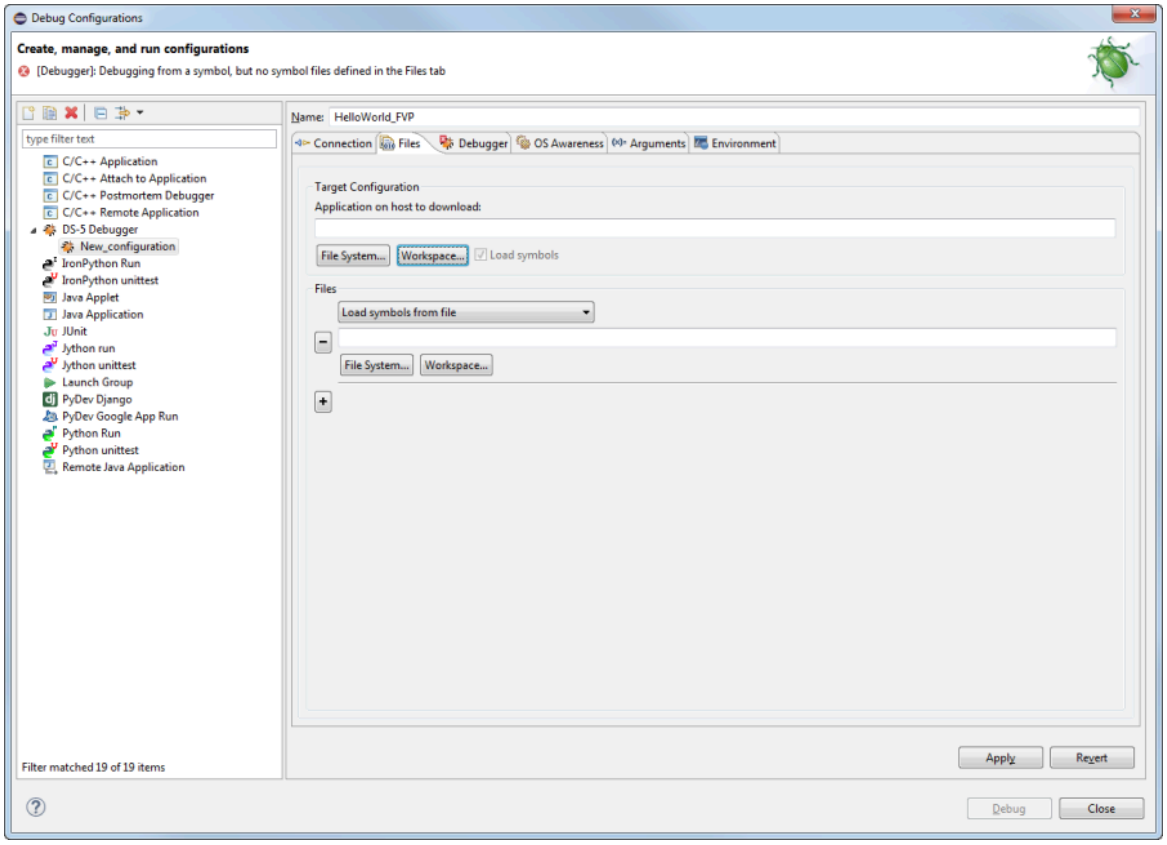
   This creates a new DS-5 debug configuration and displays the various tabs required to specify settings for loading your application on the target.



3. In the Debug Configurations dialog:
   a. Give a name to the debug configuration. For example, `HelloWorld_FVP`.
   b. In the Connection tab, under Select Target, browse and select ARM FVP (Installed with DS-5) > VE_Cortex_A9x1 > Bare Metal Debug > Debug Cortex-A9.



c. Select the Files tab, and under Target Configuration in the Application on host to download field, click Workspace.



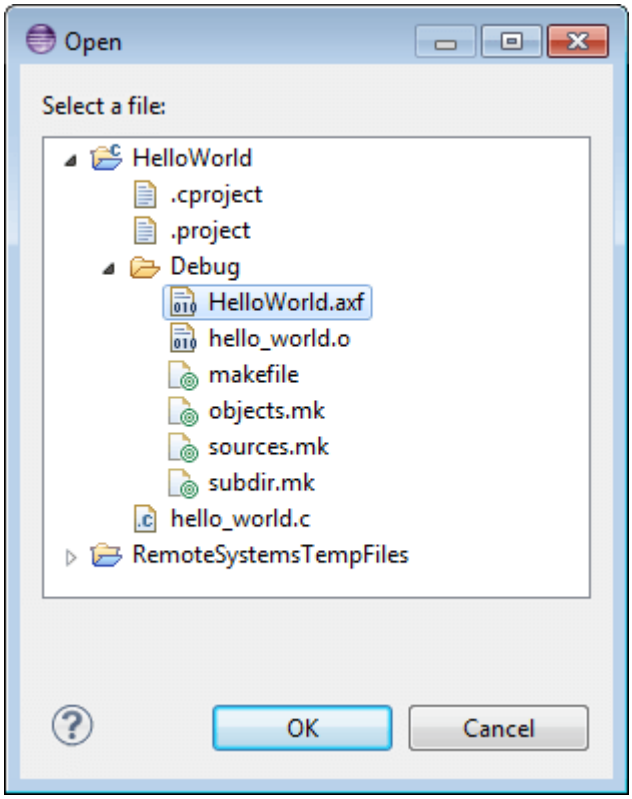
The Workspace contains the `HelloWorld.axf` application file you created when you built the Hello World project.

Note

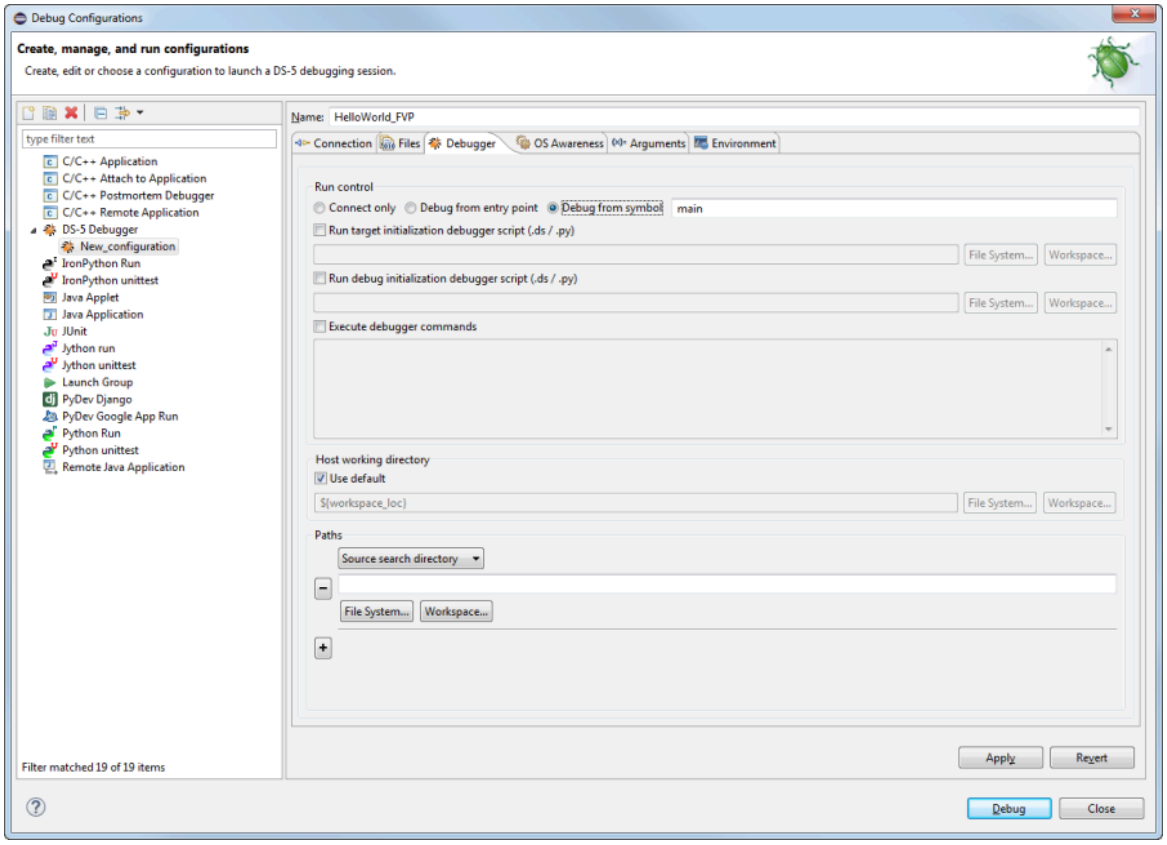
Ensure that the Load symbols option is selected.

---

d. Select `HelloWorld.axf`.

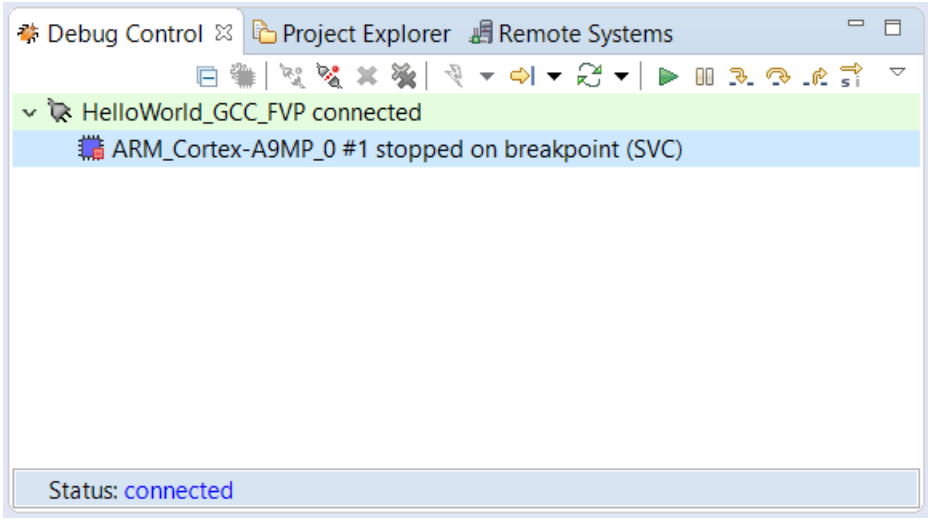


e. Select the Debugger tab, and ensure the Debug from symbol option is selected and set to `main`.



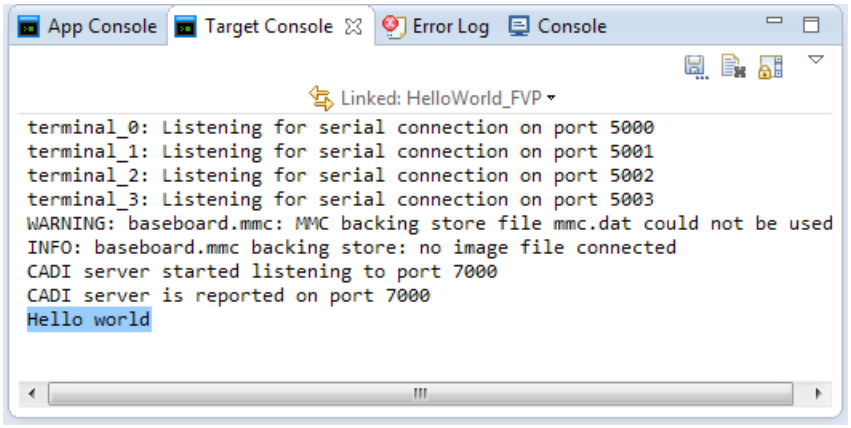
f. Click Debug to load the application on the target, and load the debug information into the debugger.

g. In the Confirm Perspective Switch dialog that appears, click Yes. DS-5 connects to the model and displays the connection status in the Debug Control view.



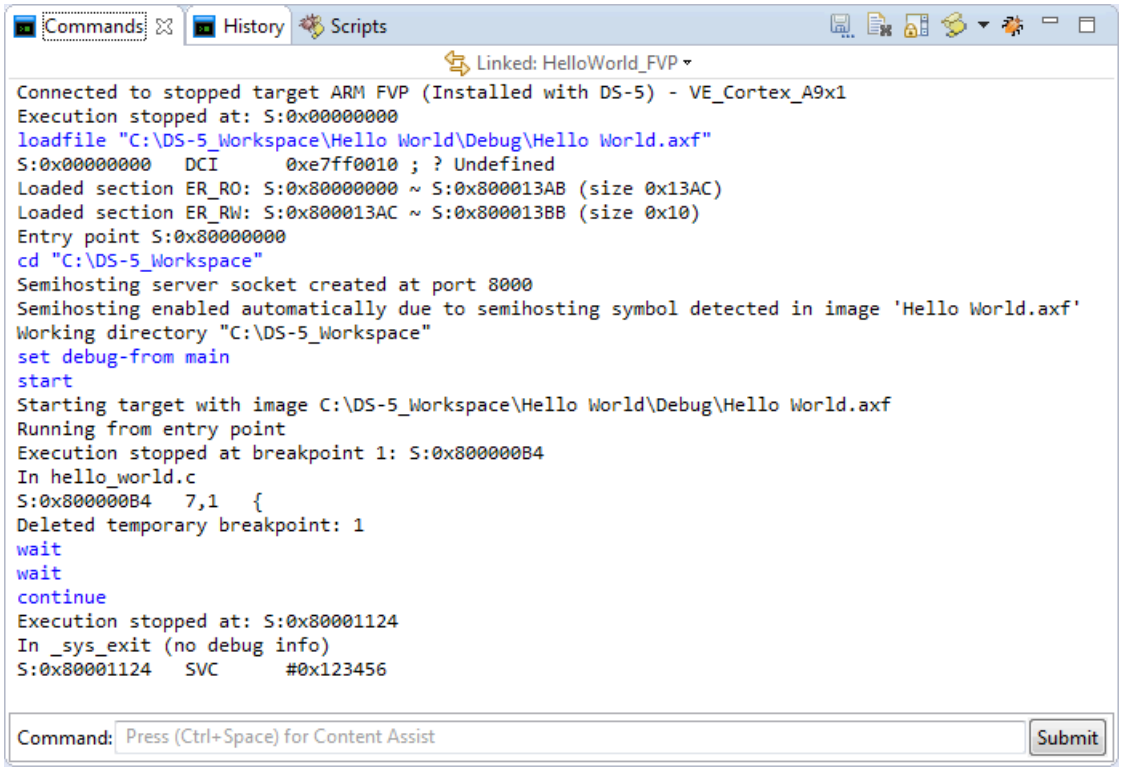
The application is loaded on the target, and has stopped at the `main()` function, ready to run.

h. Click ▶ to continue running the application. You can view the application output in the Target Console view.



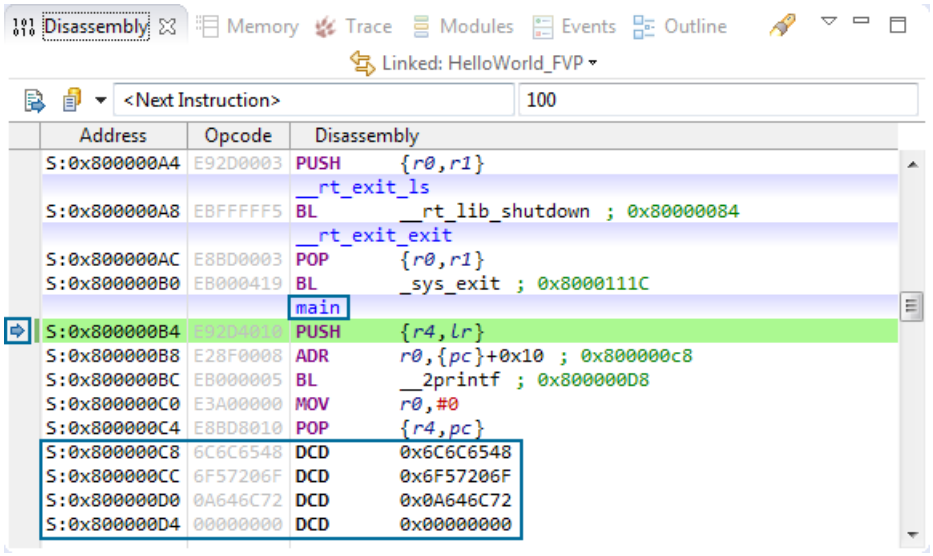
Other views display information relevant to the debug connection

- Commands view displays messages output by the debugger. Also use this view to enter DS-5 commands.



- C/C++ Editor view shows the structure of the active C, C++, or makefile. The view is updated as you edit these files.

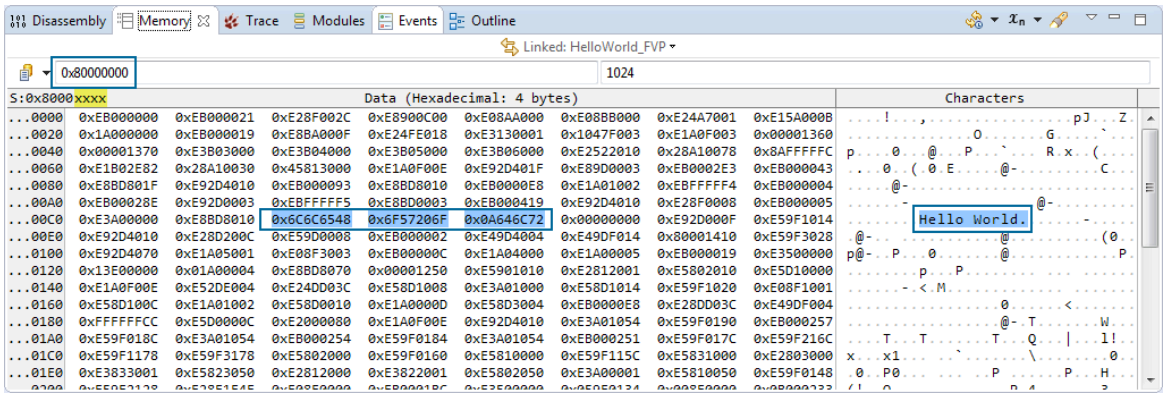
- Disassembly view shows the loaded program in memory as addresses and assembler instructions.



Indicates the location in the code where your program is stopped. In this case, it is at the `main()` function.

- Memory view shows how the code is represented in the target memory. For example, to view how the string `Hello World` from the application is represented in memory:

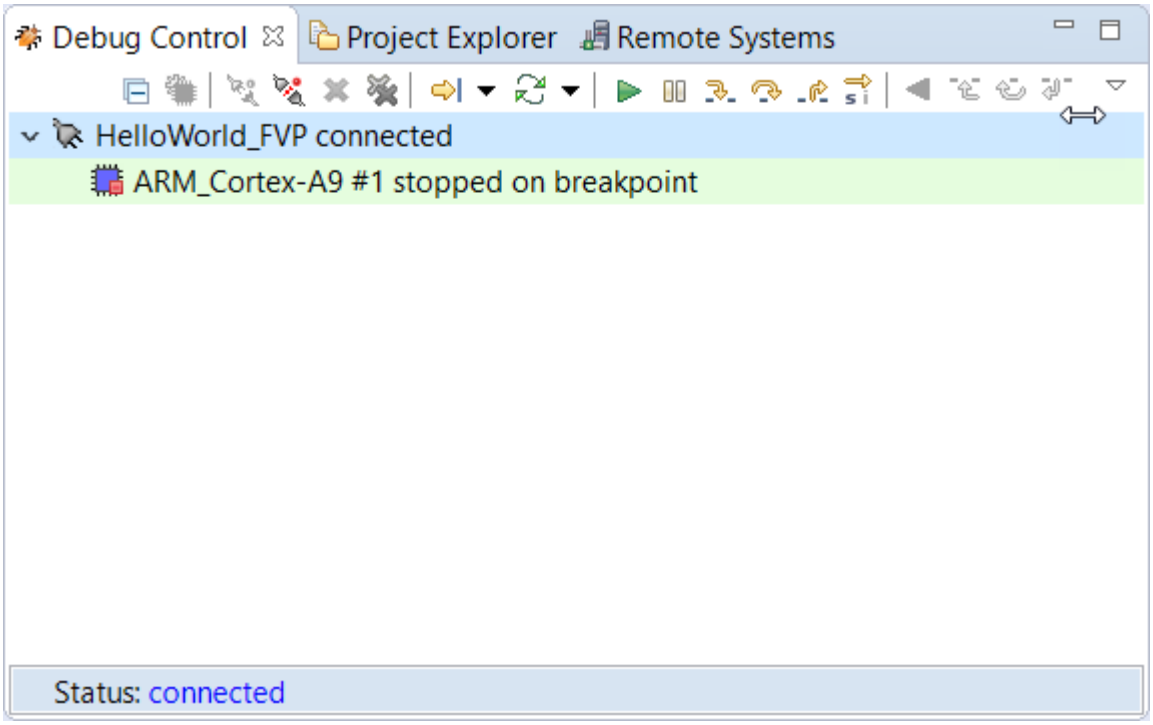
  1. Open the Memory view.
  2. In the Address field, enter `0x80000000` and press Enter on your keyboard. The view displays contents of the target's memory.
  3. Select and highlight the words `Hello World`.



In the above example, the Memory view displays the hexadecimal values for the code, and also the ASCII character equivalent of the memory values which enable you to drill down into the details of the code.

3. Step through the application

Use the controls provided in the Debug Control view to step through the application.



Click to continue processing code.

Click to interrupt or pause processing code.

Click to step through the code.

Click to step over source line.

Click to step out.

This is a toggle. Select this if you want the above controls to step through instructions.

4. Disconnect from the debug connection

To disconnect from a debug connection, you can either right-click the connection and select Disconnect from Target or select the connection and in the Debug Control view toolbar click