VECTOR

VectorCAST RSP Keil

Version 1.1

2023-08-21

Application Note AN-ACT-1-104

Author Tim Schneider

Restrictions Public Document

Abstract Information on the VectorCAST RSP Keil

Table of Contents

1 Overview 2
2 Overview 3
2.1 Use Case: Selecting the Proper Configuration 3
2.2 Use Case: Setup 7
2.3 Use Case: Troubleshooting the FVP 8
2.3.1 Migrating from FVP to VHT Simulator 8
2.4 Use Case: Debugging 9
3 Additional Resources 12
4 Trademarks 12
5 Contacts 12

1 Overview

VectorCAST Runtime Support Package (RSP) is an extension of the VectorCAST product. It provides an interface layer that allows you to use the VectorCAST testing techniques and methods on an embedded target processor. The VectorCAST RSP will always run on your host platform (the same platform as the compiler). Only the VectorCAST generated test harness will be downloaded to, and executed on, the embedded target.

A VectorCAST RSP is always customized to a particular Target CPU, Cross Compiler, and Run-Time environment (or kernel). As such, the information presented here contains sections for this specific RSP.

This application note provides details on configuring VectorCAST to run tests on an Arm Clang KEIL target. The target can be a live board, or a simulated board built into Keil's IDE, µVision 5 (UV5).

VectorCAST uses the Arm Clang compiler and linker to build the VectorCAST test harness into an UV5 application. The application is then transferred to the Keil target for execution directly via UV5's command line interface and IDE.

VectorCAST provides off the shelf Arm Clang Keil RSPs for the ARM architecture. If you need support for a different architecture or run into any issues configuring VectorCAST, please contact support@vector.com

2 Overview

2.1 Use Case: Selecting the Proper Configuration

The first thing to do when configuring your Arm Clang Keil RSP is to select the ARM architecture as seen in Figure 1.

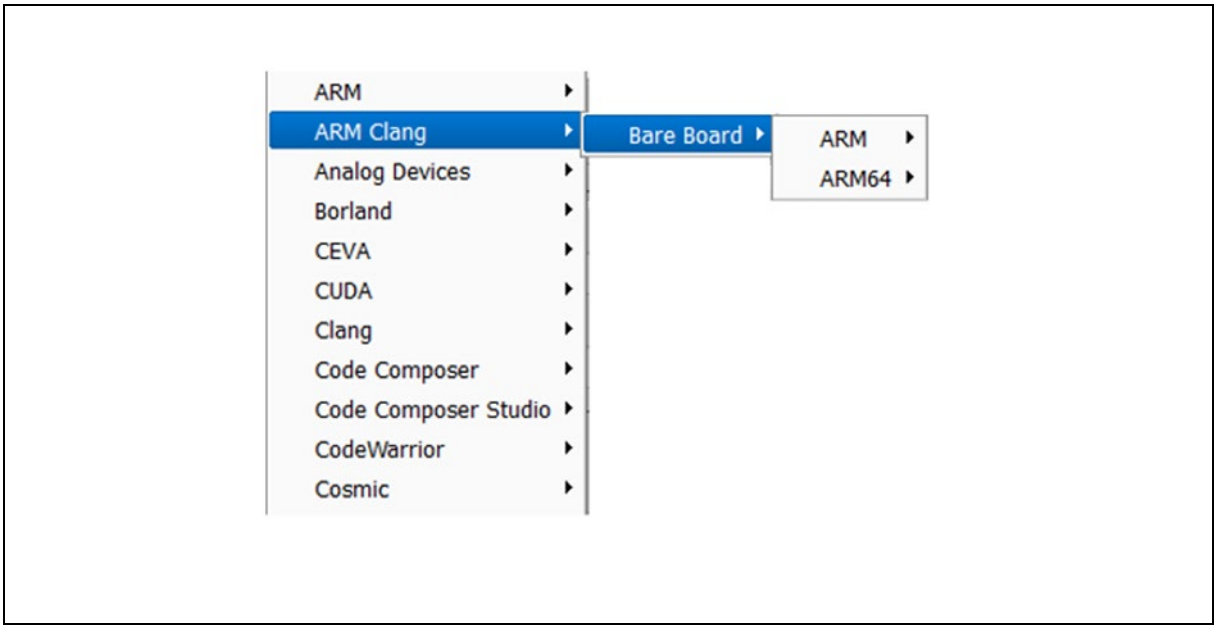


Figure 1 – Selecting the ARM Configuration

The second step is to select the correct Arm Cortex CPU as your target. Arm Clang supports a multitude of CPUs that are not being utilized by Keil. Select one of the “Cortex-M” series CPUs within the list since they are supported directly by Keil.

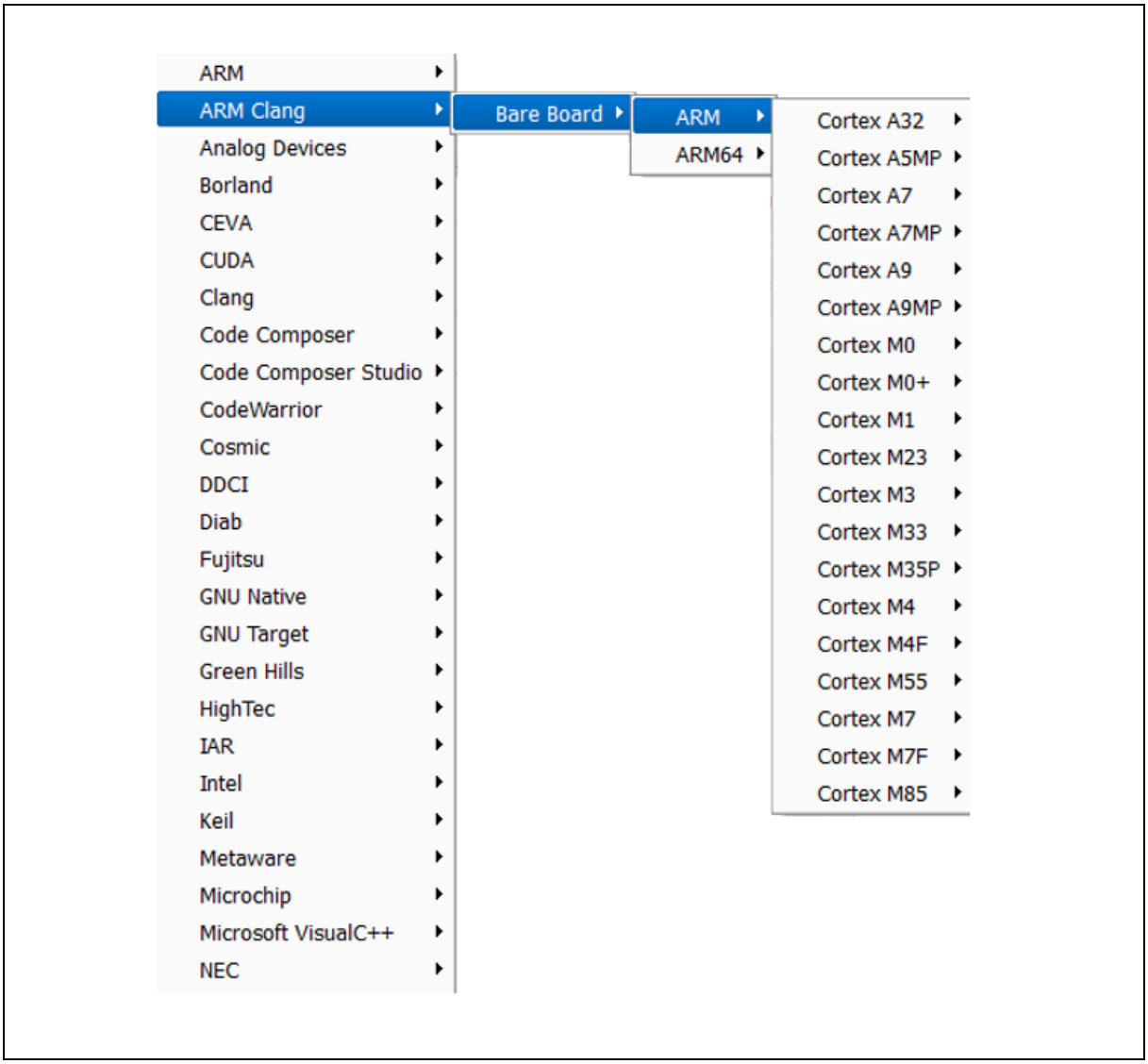


Figure 2 – Selecting the Cortex Configuration

Simply select the Keil Simulator from the next drop down. While some of the CPUs will require an FVP simulator to work correctly, the Keil Simulator must be chosen to utilize the UV5 IDE and the Keil configuration. Instructions for running tests on the FVP simulator outside of Keil is provided in a separate App Note that you can find on the Vector website.

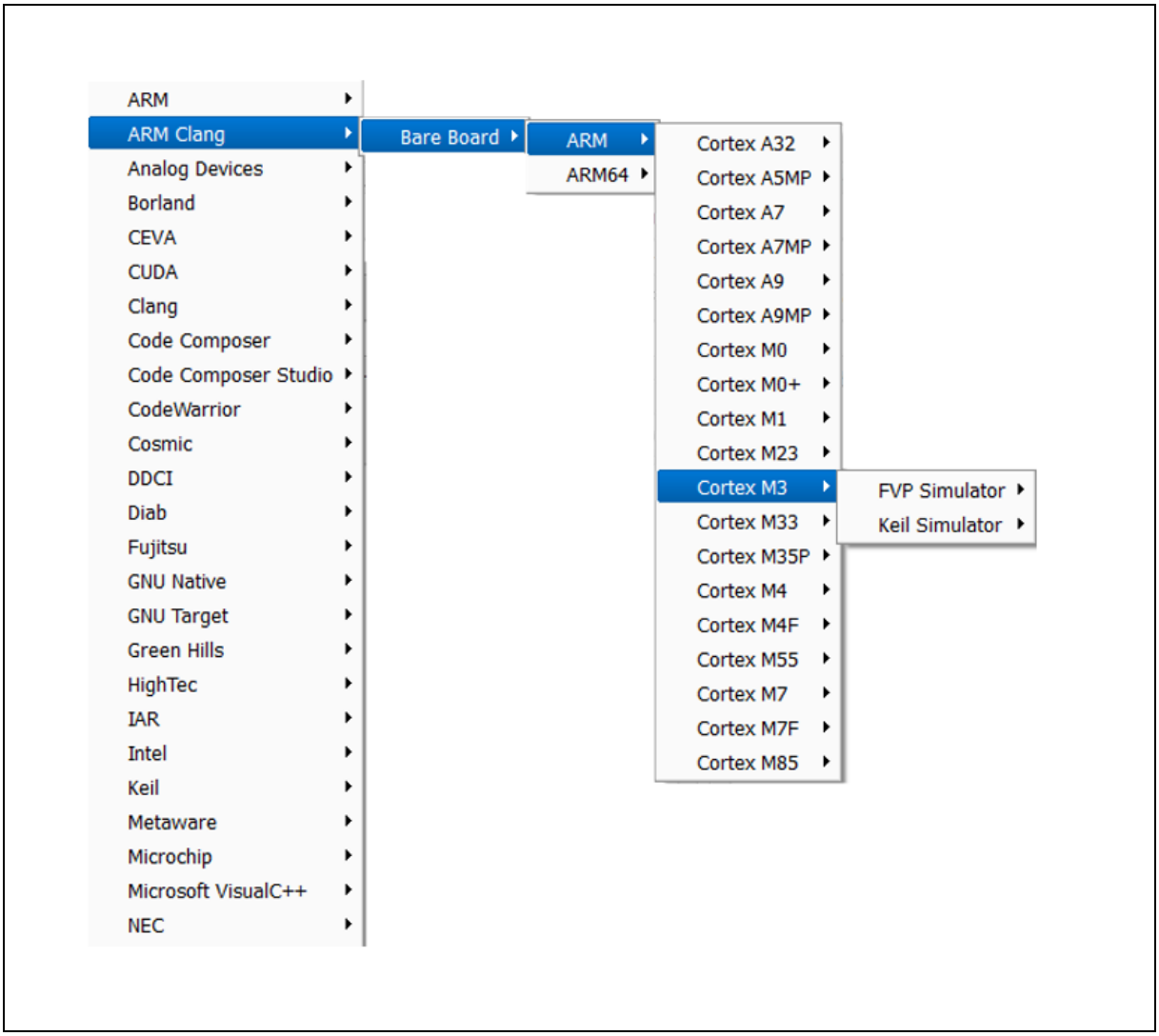


Figure 3 – Selecting the Simulator Configuration

The last step is to select the type of code to test. Choose C for only testing C code, choose C++ for either C++ or a combination of C and C++.

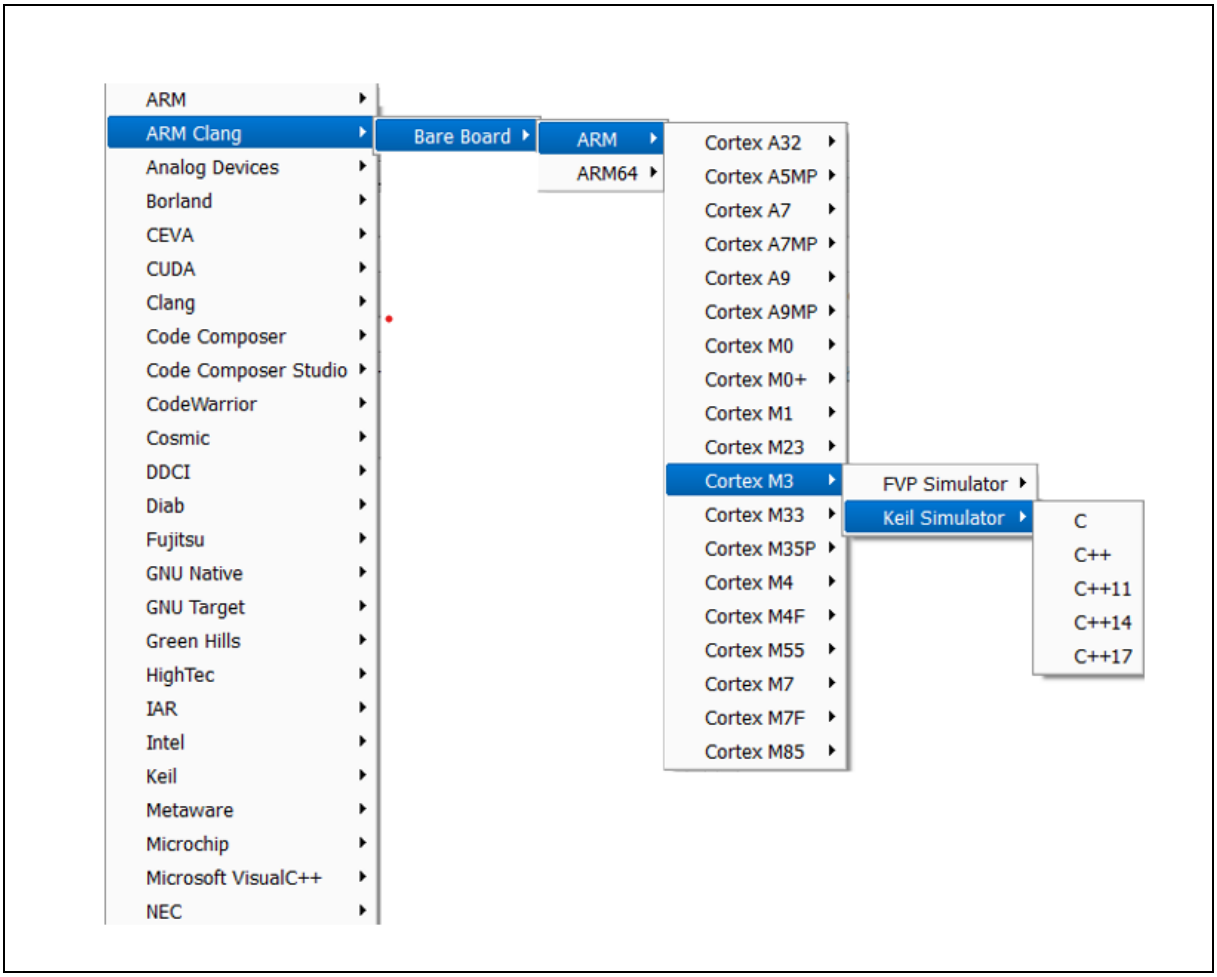


Figure 4 – Selecting the Coding Configuration

Once selected either C or any C++ version, the configuration will show up in your project and the compiler node will be populated with the proper data, seen in Figure 5.

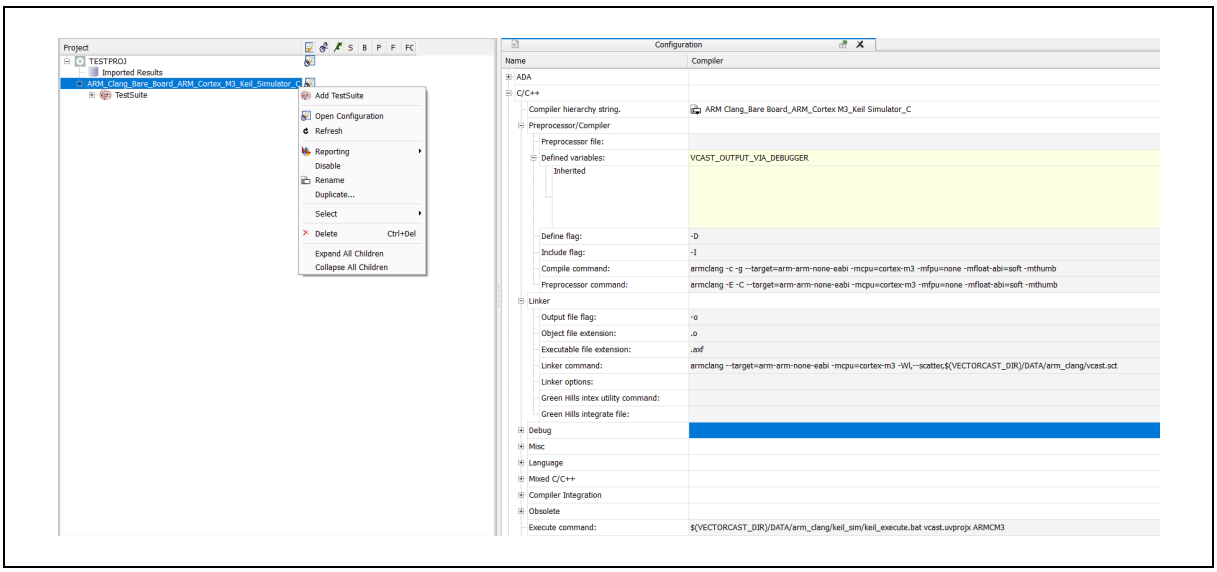


Figure 5 – Project Configuration

2.2 Use Case: Setup

The Keil development environment must be setup before launching VectorCAST so that VectorCAST has access to the compiler, linker, debugger, etc.

VectorCAST uses µVision 5 (UV5) to debug the test harness and requires a path to the UV5 IDE to be added to the path environment variable.

A Fixed Virtual Platform (FVP) simulator within the Keil installation will have to be included if using one of the following CPUs: Cortex-M23, Cortex-M33, Cortex-M35P, Cortex-M55, Cortex-M85. The FVP simulator that is recommended by ARM is the ARM Ecosystem FVP Corstone-300 machine, which at the time of writing, can be found at https://developer.arm.com/downloads/-/arm-ecosystem-fvps. There is a video embedded on the website highlighting the installation instructions. Once it is installed, it must be placed within the local Keil installation, for example, `D:\Keil_v5\ARM\FVP\Corstone-300`. This file path will not exist natively within Keil installation and must be created exactly. The UV5 will recognize this location automatically within the provided project file.

VectorCAST recommends putting the setting of the path to the UV5 IDE and the path to the Arm Clang compiler into a startup script called **setup_VectorCAST.bat** or **setup_VectorCAST.sh** depending on if using Windows or Linux.

setup_VectorCAST.bat script will end up looking something like this:

```
set path=D:\Keil_v5\UV4;D:\Keil_v5\ARM\ARMCLANG\bin;%path%
```

The **setup_VectorCAST.bat** script will need to be called before execution of any VectorCAST CLI commands (such as `%VECTORCAST_DIR%\manage` or `%VECTORCAST_DIR%\clicast`).

The **setup_VectorCAST.bat** script will also need to be called before launching the VectorCAST GUI. VectorCAST recommends creating another script called **start_VectorCAST.bat** that will call **setup_VectorCAST.bat** and then launch the VectorCAST GUI.

start_VectorCAST.bat will look like this:

```
call %~dp0setup_VectorCAST.bat
start %VECTORCAST_DIR%\vcastqt
```

The `%~dp0` is a windows batch syntax that will expand the directory path so that **`setup_VectorCAST.bat`** will be called from the same directory that **start_VectorCAST.bat** exists in. This creates a shortcut to **start_VectorCAST.bat** and if **setup_VectorCAST.bat** is in the same directory as **start_VectorCAST.bat** then the shortcut will work.

Use **start_VectorCAST.bat** to launch VectorCAST to be able to build or execute unit test environments.

2.3 Use Case: Troubleshooting the FVP

Ensure that the FVP is placed within the proper location inside the local Keil installation. UV5 uses the environment variable **$KARM** to store the location of the ARM directory within the local Keil installation. If the path to the FVP does not match the exact path that UV5 uses within the prefabricated project ($KARM/FVP/Corstone-300/…), the error below will be thrown during build/execution within UV5.

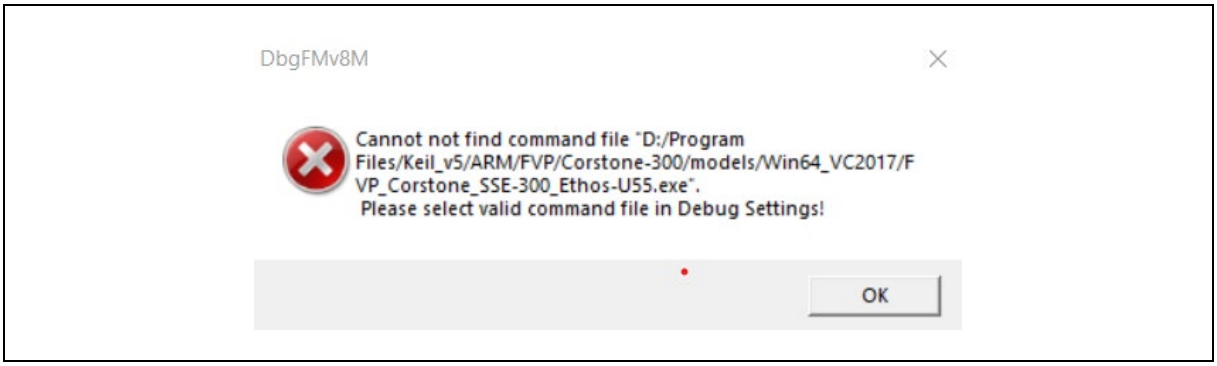


Figure 6 – Error Message

Ensure that this path is created exactly as written so it can be recognized when running the FVP simulator. **$KARM** eliminates the need for the user to use the exact path because the environment variable recognizes where `~/Keil_v5/ARM/` is located. Therefore, everything after `~/ARM/` must be created by the user.

Once this error is thrown, simply click out of the rest of the error messages and close UV5 and return to VectorCAST. At this point, the testcase/environment build will have failed. Resolve the pathing issue and try again, this will fix the issue.

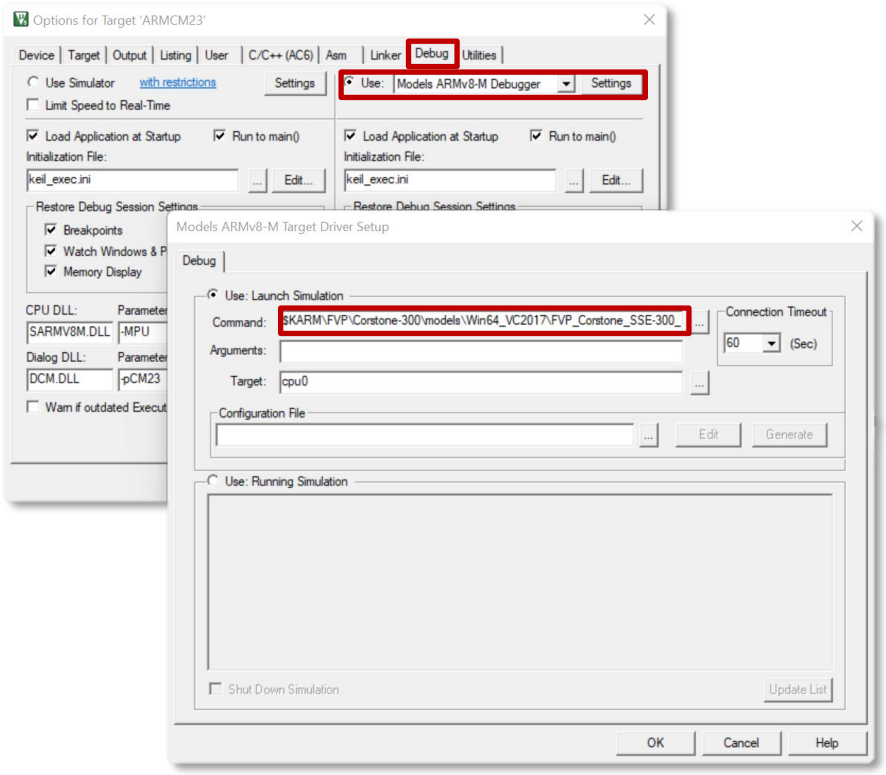
2.3.1 Migrating from FVP to VHT Simulator

If you are using Keil MDK version 5.38 or higher, the FVP simulator has been replaced with the VHT simulator in the Keil installation. You will need to update the project as you did in Section 2.3. This time instead of finding the path to the FVP simulator, you will need to find the path to the VHT simulator. Typical installations place the VHT simulator under **$KARM/VHT.**

The change could be a simple as this:

FROM: $KARM\FVP\Corstone-300\models\Win64_VC2017\FVP_Corstone_SSE-300_Ethos-U55.exe
TO: $KARM\VHT\VHT_Corstone_SSE-300_Ethos-U55.exe

The change would go int the vcast Keil MDK project that VectorCAST uses to run the unit tests in the Options > Debug > Model ARMv8M Debugger > Settings > Command dialog.



Once you have made these changes, you should store the project outside the VectorCAST Installation directory and add it to the list of environment files in Tools > Options > C/C++ > Miscellaneous > Environment files to be copied in.

In addition, by executing directly on the VHT simulator instead of going through the Keil MDK debugger, the execution time should be significantly decreased. if you want to execute via the VHT simulator directly, the user will need to do the following:

1. If the user has added the path to the FVP simulator to their default path, remove it and replace with the path to the VHT simulator and restart VectorCAST:
2. Modify the Execute command in Tools > Options > C/C++:

   FROM: $(VECTORCAST_DIR)/DATA/arm_clang/keil_sim/keil_execute.bat vcast.uvprojx ARMCM23
   TO: VHT_MPS2_Cortex-M23 -C fvp_mps2.DISABLE_GATING=1

   *These changes shown above are for the Cortex-M23 specifically

3. Change the IO by
   a. Removing the defined variable VCAST_OUTPUT_VIA_DEBUGGER
   b. Unchecking Buffer IO from Tools > Options > Target > Input/Output Options
4. Rebuilding the environment.

2.4 Use Case: Debugging

Debugging VectorCAST testcases is done using the UV5 IDE.

The following steps will describe the process for debugging a testcase from the UV5 IDE.

When right clicking on a testcase and select ‘Execute with Debug’ the first thing VectorCAST will do is create a **keil_debug.ini** script. This is a script that will be used from UV5 to start the debug instance and create breakpoints within the test harness. After creation of the **keil_debug.ini** script, VectorCAST will launch UV5 via a batch script **keil_debug.bat**.

VectorCAST users will not need to setup a VectorCAST debug configuration in UV5. The UV5 project comes with your install of VectorCAST, and setup based upon the CPU selected within VectorCAST.

The following are steps to run the debug instance within UV5 IDE.

Once UV5 has been launched, the test harness will run automatically to a breakpoint set at the function under test and the code can be debugged.

If running the code via the debugger manually, go to Debug -> Run… and this will run the test harness to completion, see Figure 7.

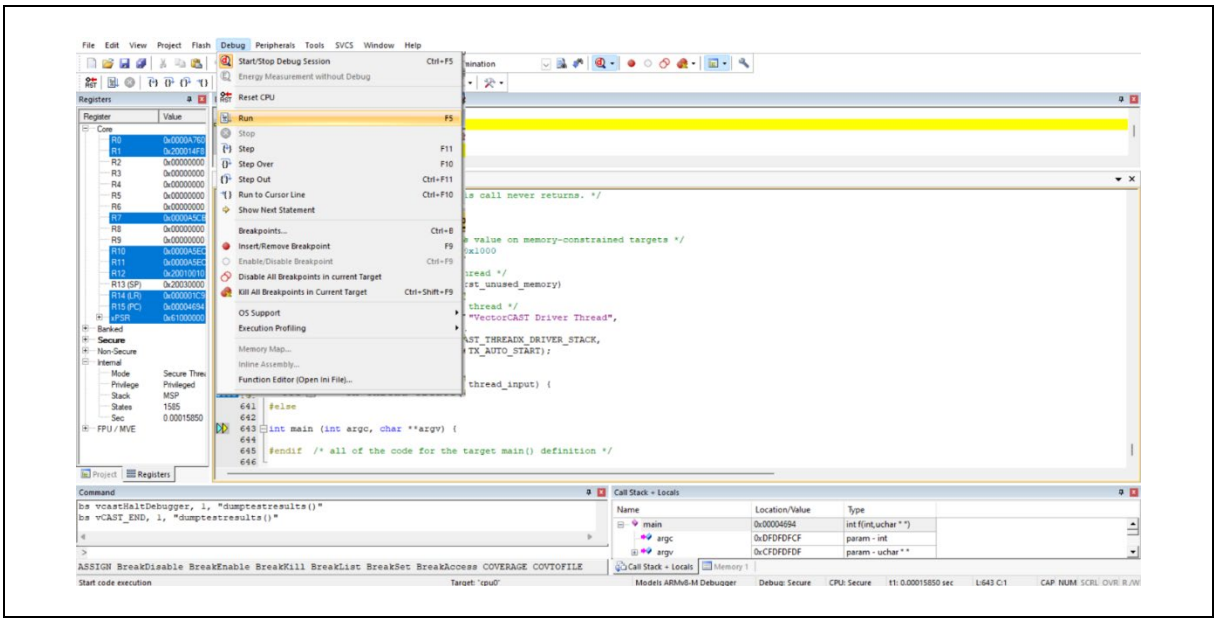


Figure 7 – Debugging and Running Manually

Once debugging is finished, make sure that test harness has finished running to completion. This will be indicated in the command box as **VCAST.END:END** as shown below.

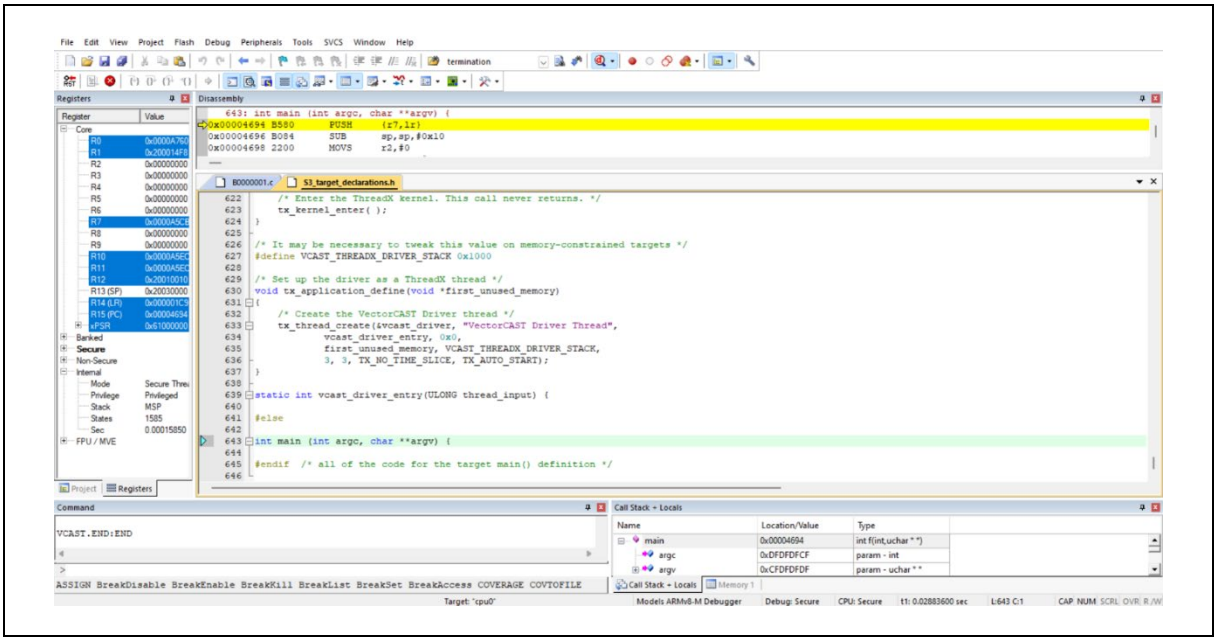


Figure 8 – Test Harness

Simply close out the UV5 IDE and VectorCAST will recognize the test as completed.

3 Additional Resources

https://www2.keil.com/mdk5

4 Trademarks

All mentioned names are either registered or unregistered trademarks of their respective owners.

5 Contacts

For a full list with all Vector locations and addresses worldwide, please visit http://vector.com/contact/.