

VectorCAST RSP RISC-V

Version 1.2

2025-04-01

Application Note AN-ACT-1-109

Author	Derek Opitz
Restrictions	Public Document
Abstract	Information on the VectorCAST RSP with RISC-V

Table of Contents

1	Overview	2
2	Setup	3
2.1	Use Case: Selecting the Proper Configuration	3
2.2	Use Case: Environment Setup.....	3
2.3	Use Case: Building Environments	5
2.4	Use Case: Testcase Execution.....	5
2.5	Use Case: Debugging	6
3	Additional Resources	6
4	Trademarks	6
5	Contacts	6

1 Overview

VectorCAST Runtime Support Package (RSP) is an extension of the VectorCAST product. It provides an interface layer that allows the customer 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 a RISC-V processor using the RISC-V GNU compiler. The target can be a live board, or a simulated board using the renode emulator.

VectorCAST uses the RISC-V GNU compiler and linker to build the VectorCAST test harness into an executable image that can be loaded onto a RISC-V processor or renode emulator. VectorCAST then uses the gdb debugger connection to load the executable image onto the RISC-V processor or renode emulator and run the test and collect the test results.

VectorCAST provides off the shelf RSPs for RISC-V rv32 and rv64. The rv32 support is generic and executes on the renode emulator, the rv64 support is for the Microchip PolarFire SoC Icicle Kit and supports test execution on both the Icicle board as well as the renode emulator. For help configuring VectorCAST for a different processor or debugger please contact support@vector.com.

2 Setup

2.1 Use Case: Selecting the Proper Configuration

The first thing to do when configuring RISC-V RSP is to select the correct configuration Figure 1. If your architecture is rv32, choose RISCV32. If your architecture is rv64, choose RISCV64. Then choose whether you want to test on the Renode Simulator or on the Icicle hardware. Then, Choose C for only testing C code, choose C++ for either C++ or a combination of C and C++.

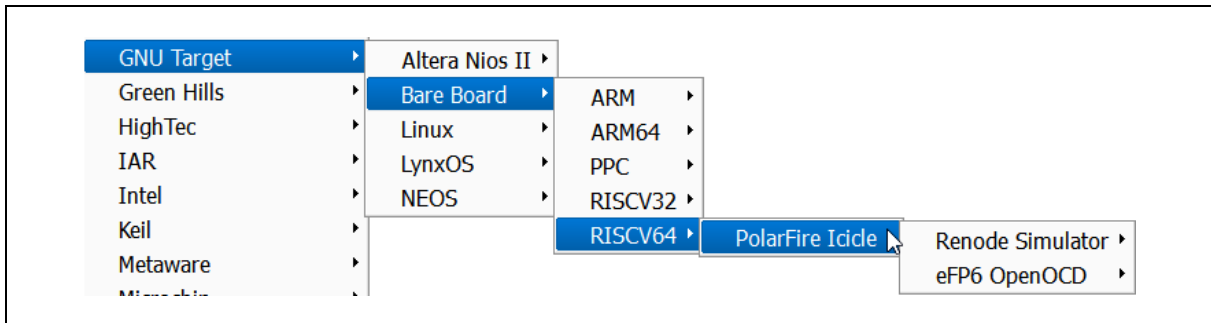


Figure 1 – Selecting the RISC-V Configuration

Once selected either C or any C++ version, the configuration will show up in the project and the compiler node will be populated with the proper data, seen in Figure 2.

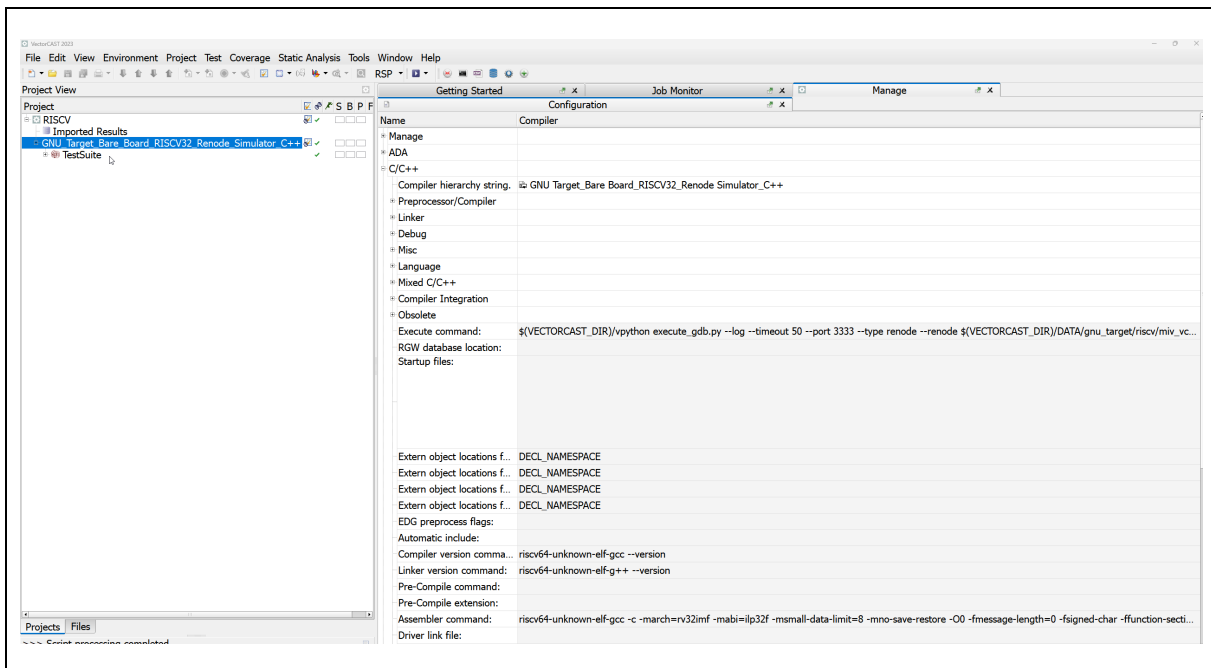


Figure 2 – Project Configuration

2.2 Use Case: Environment Setup

The path to the RISC-V gnu compiler and the renode emulator need to be on the path environment variable.

VectorCAST recommends putting the path into a startup script called **setup_VectorCAST.bat** or **setup_VectorCAST.sh** depending on if using Windows or Linux.

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

```
set SOFTCONSOLE_INSTALL=D:\Microchip\SoftConsole-v2022.2-RISC-V-747
set path=%SOFTCONSOLE_INSTALL%\renode\bin;%SOFTCONSOLE_INSTALL%\riscv-
unknown-elf-gcc\bin;%path%
```

setup_VectorCAST.sh script for Linux will end up looking something like this:

```
export SOFTCONSOLE_INSTALL=/opt/Microchip/SoftConsole-v2022.2-RISC-V-747
export PATH=$SOFTCONSOLE_INSTALL/renode/bin:$SOFTCONSOLE_INSTALL/riscv-
unknown-elf-gcc/bin:$PATH
```

The first line of **setup_VectorCAST** script sets the environment variable **SOFTCONSOLE_INSTALL**, the second line puts the path to the compiler and the path to the renode emulator on the environment path so that VectorCAST can call the compiler and the emulator.

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

start_VectorCAST.bat for Windows will look like this:

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

start_VectorCAST.sh for Linux will look like this:

```
source ./setup_VectorCAST.sh
$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 allows for a shortcut to **start_VectorCAST.bat** to be used and if **setup_VectorCAST.bat** is in the same directory as **start_VectorCAST.bat** then the shortcut will work. The Linux equivalent is the `./` present in front of **setup_VectorCAST.sh**.

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

2.3 Use Case: Building Environments

Following the setup instructions in this document should allow the building of VectorCAST environments for the RISC-V processor. If still encountering difficulties in building environments for the RISC-V processor, contact support@vector.com.

2.4 Use Case: Testcase Execution

Instructions in this document should be enough to execute testcases on the renode emulator without any further modifications. If testing on the Icicle hardware, then you will need to start the OpenOCD server using the command below. The path to openocd.exe will need to be on your system path, or you will need to give the full path to openocd.exe. The path to the Microsemi-riscv.cfg file may need to be changed to match your SoftConsole installation location.

Windows:

```
openocd.exe --command "set DEVICE MPFS" --file "C:\Microchip\SoftConsole-
v2022.2-RISC-V-747\openocd\share\openocd\scripts\board\microsemi-
riscv.cfg"
```

Linux:

```
openocd --command "set DEVICE MPFS" --file "/opt/Microchip/SoftConsole-
v2022.2-RISC-V-747/openocd/share/openocd/scripts/board/microsemi-
riscv.cfg"
```

If you are executing on a different target, then additional modifications may be required. Please contact support@vector.com for more details.

VectorCAST uses the Execute command in the VectorCAST options to execute tests. For the RISC-V RSP, VectorCAST uses a python script to launch the renode emulator and connect to the emulator via gdb. A gdb script is used to load the executable into the emulated processor and collect the test results.

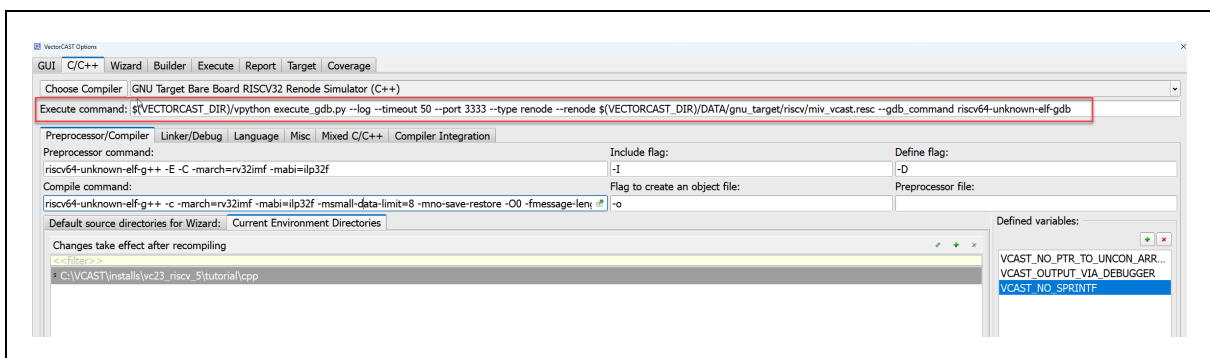
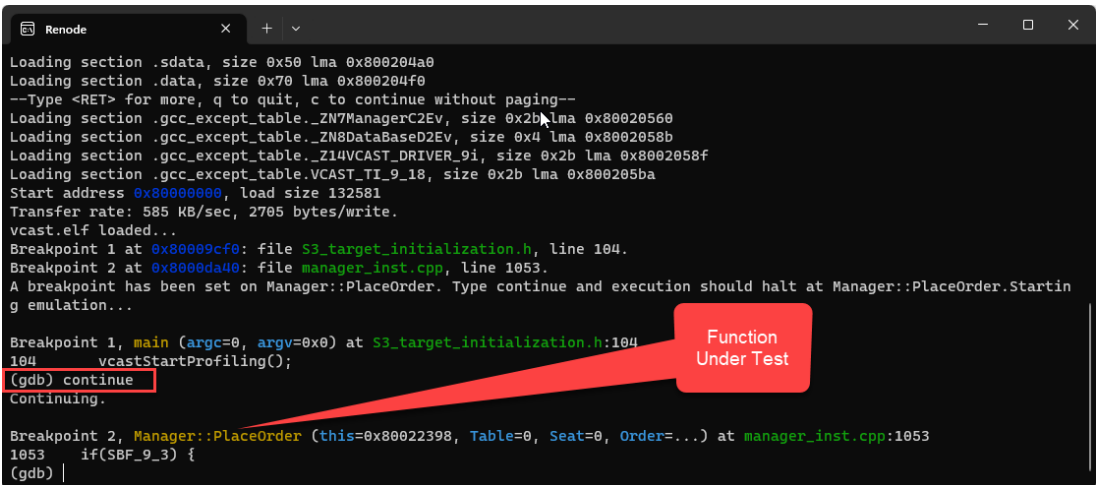


Figure 3 – Testcase Execution

2.5 Use Case: Debugging

Debugging VectorCAST testcases is done using gdb.

When right clicking on a testcase and select 'Execute with Debug' VectorCAST will launch the renode emulator, then launch gdb. VectorCAST will set a breakpoint on the function under test. Type continue at the gdb prompt and execution should stop at the function under test.



```
Renode
Loading section .sdata, size 0x50 lma 0x800204a0
Loading section .data, size 0x70 lma 0x800204f0
--Type <RET> for more, q to quit, c to continue without paging--
Loading section .gcc_except_table._ZN7ManagerC2Ev, size 0x2b lma 0x80020560
Loading section .gcc_except_table._ZN8DataBaseD2Ev, size 0x4 lma 0x8002058b
Loading section .gcc_except_table._Z14VCAST_DRIVER_9i, size 0x2b lma 0x8002058f
Loading section .gcc_except_table.VCAST_TI_9_18, size 0x2b lma 0x800205ba
Start address 0x80000000, load size 132581
Transfer rate: 585 KB/sec, 2705 bytes/write.
vcast.elf loaded...
Breakpoint 1 at 0x80009cf0: file S3_target_initialization.h, line 104.
Breakpoint 2 at 0x8000da40: file manager_inst.cpp, line 1053.
A breakpoint has been set on Manager::PlaceOrder. Type continue and execution should halt at Manager::PlaceOrder.Startin
g emulation...

Breakpoint 1, main (argc=0, argv=0x0) at S3_target_initialization.h:104
104 vcastStartProfiling();
(gdb) continue
Continuing.

Breakpoint 2, Manager::PlaceOrder (this=0x80022398, Table=0, Seat=0, Order=...) at manager_inst.cpp:1053
1053 if(SBF_9_3) {
(gdb) |
```

Figure 4 – Execute with Debug

The SoftConsole IDE may also be used for debugging the VectorCAST testcases.

3 Additional Resources

<https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-design-tools/soc-fpga/softconsole>

[PolarFire Soc FPGAs](#)

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/>.