

VectorCAST RSP Microchip

Version 1.0

2023-09-01

Application Note AN-ACT-1-110

Author	Derek Opitz
Restrictions	Public Document
Abstract	Information on the VectorCAST RSP Microchip

Table of Contents

1	Overview	2
2	Setup	3
2.1	Use Case: Selecting the Proper Configuration.....	3
2.2	Use Case: Environment Setup	4
2.3	Use Case: Building Environments.....	5
2.4	Use Case: Testcase Execution	5
2.5	Setting up VectorCAST for target execution.....	6
2.6	Known mdb issues	6
2.7	Use Case: Debugging	7
3	Additional Resources	10
4	Trademarks	10
5	Contacts.....	10

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 a Microchip processor using the xc8, xc16, or xc32 compiler. The target can be a live board or the Microchip simulator.

VectorCAST uses the xc8, xc16, or xc32 compiler and linker to build the VectorCAST test harness into an executable image that can be loaded onto a Microchip processor or simulator. VectorCAST then uses the mdb command line debugger to load the executable image onto the Microchip processor or simulator and run the test and collect the test results.

VectorCAST provides off the shelf RSPs for the Microchip PIC32, PIC24, dsPIC30, dsPIC33, and PIC18 processors. If you need support for a different architecture, need help configuring VectorCAST for your processor or debugger, or run into any issues configuring VectorCAST, please contact support@vector.com.

2 Setup

2.1 Use Case: Selecting the Proper Configuration

The first thing to do when configuring your Microchip RSP is to select the correct Microchip compiler as shown in Figure 1. Choose XC32 if your processor is a PIC32. Choose XC16 if your processor is a PIC24, dsPIC30, or dsPIC33. Choose XC8 if your processor is a PIC18.

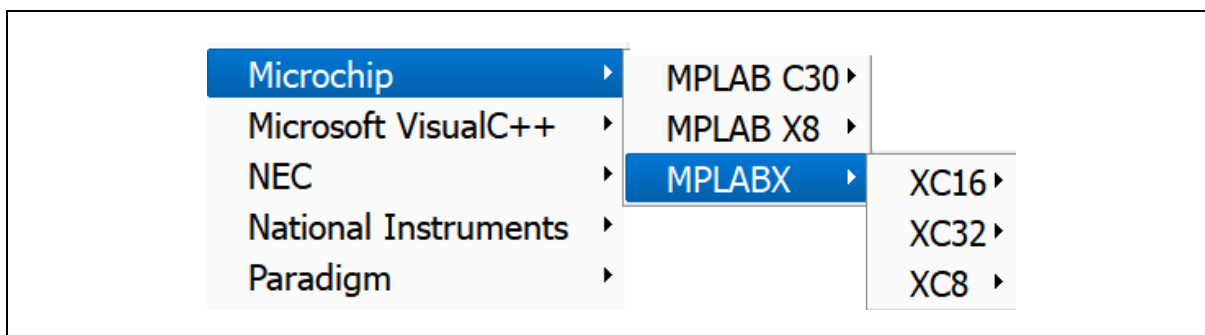


Figure 1 – Selecting the Microchip compiler

Once you have selected the compiler, the next step is to select the correct debugger. If you want to run on the Microchip simulator, choose Simulator. If you want to run on target hardware select your debugger. If you are using an ICD4 debugger, then choose ICD3. If you are using another debugger that is not listed, contact support@vector.com and we can help configure VectorCAST for that debugger.

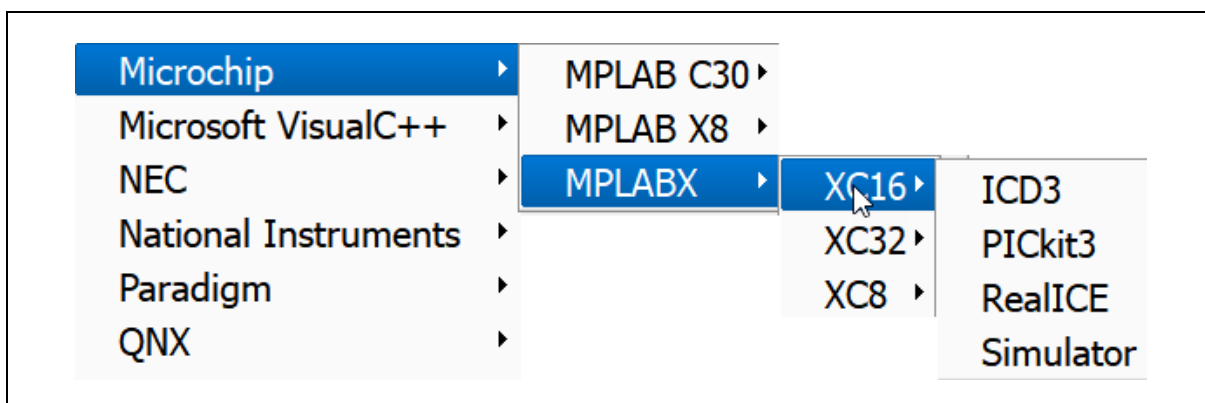


Figure 2 – Selecting the Microchip debugger

Once you have selected the debugger, the configuration will show up in your project and the compiler node will be populated with the proper data, seen in Figure 3.

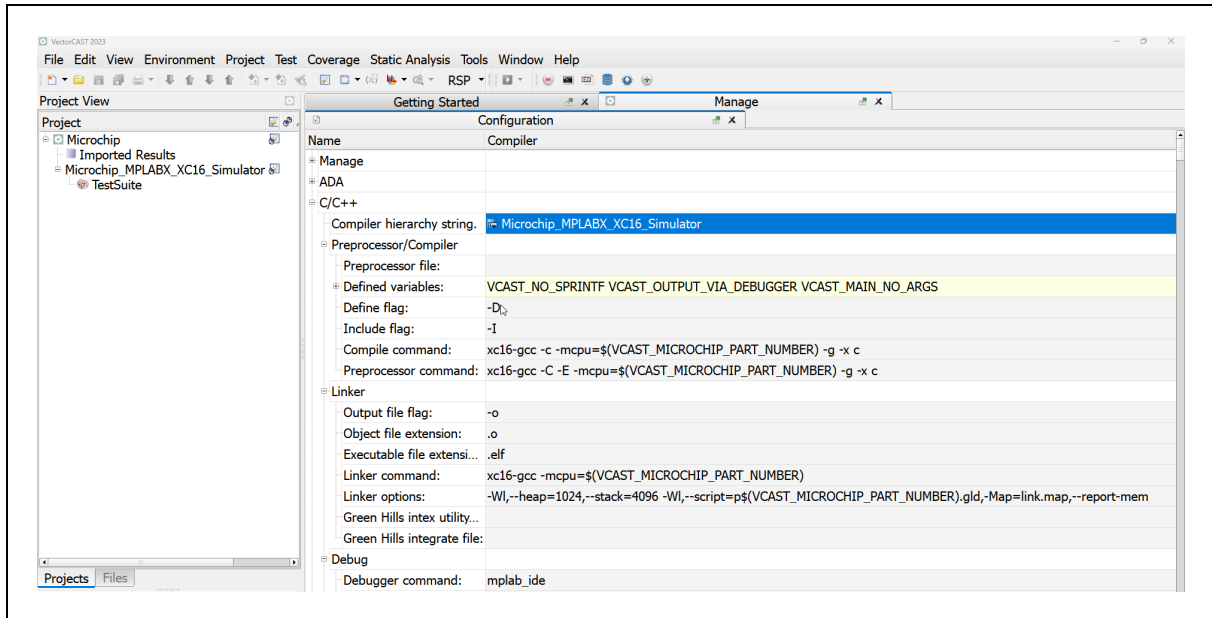


Figure 3 – Project Configuration

2.2 Use Case: Environment Setup

VectorCAST needs to know the exact processor that is being used for testing. VectorCAST determines this using the environment variable `VCAST_MICROCHIP_PART_NUMBER`. Additionally, VECTORCAST needs to know where your compiler is located. VectorCAST determines this using one of the following environment variables (depending on which compiler you chose in your configuration): `VCAST_XC8_INSTALL_DIR`, `VCAST_XC16_INSTALL_DIR`, or `VCAST_XC32_INSTALL_DIR`. Additionally, the path to the compiler, the mdb debugger, and the MPLAB X executable need to be on your path.

VectorCAST recommends putting `VCAST_MICROCHIP_PART_NUMBER`, the compiler env var, and 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 will end up looking something like this:

```
set VCAST_MICROCHIP_PART_NUMBER=33EP32MC504
set VCAST_XC16_INSTALL_DIR=G:\ProgramFiles\Microchip\xc16\v2.10
set
path=%VCAST_XC16_INSTALL_DIR%\bin;%VCAST_XC16_INSTALL_DIR%\..\..\MPLABX\v5.30\mplab_platform\bin;%path%
```

The first line of **setup_VectorCAST.bat** sets the environment variable `VCAST_MICROCHIP_PART_NUMBER`, the second line sets the compiler env var (in this case for XC16 compiler), and the third line puts the path to the compiler and the path to mdb on the environment path so that VectorCAST can call the compiler and the debugger.

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 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.

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

2.3 Use Case: Building Environments

If you followed the setup instructions in this App Note, then you should be able to build VectorCAST environments for your Microchip processor. It's possible with the 8-bit and 16-bit processors that the default VectorCAST harness may be too large to fit into the available flash and ram space. There are many options inside VectorCAST that will reduce the harness size to allow VectorCAST to run on resource limited processors. If you need assistance reducing the harness size or are having any other issues building environments for your Microchip processor, please contact support@vector.com.

2.4 Use Case: Testcase Execution

If you followed the setup instructions in this App Note, then you should be able to execute testcases on the Microchip simulator without any further modifications. If you need to execute testcases on a live target, then some modifications are required. The modifications for a live target will be addressed in the next section. If you have followed this App Note and are still having issues executing testcases, please contact support@vector.com.

VectorCAST uses the Execute command in the VectorCAST options to execute tests. For the Microchip RSP, VectorCAST uses a batch script to setup environment, then calls a python script to launch mdb and run the test case. A mdb execution script is used to load the executable onto the processor and collect the test results.

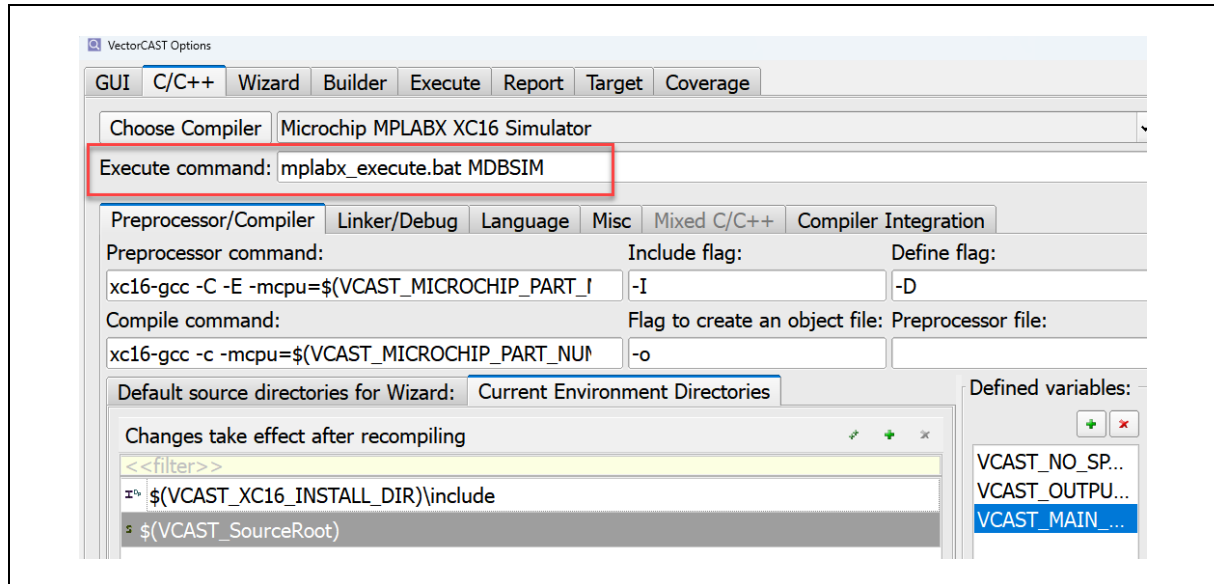


Figure 4 – Testcase Execution

2.5 Setting up VectorCAST for target execution

If you are using the ICD4 debugger, the only change required is to change ICD3 to ICD4 in the Execute command.

To run on a live target, the config words for that target must be configured correctly. VectorCAST provides a source file in the install that can be modified to add the config words. The source file is located in the VectorCAST install under DATA/microchip/mplabx/main.c. There are example config words for the different processors, but the unique config words for your processor will need to be added.

In the config words, verify that the watchdog timer is disabled: `#pragma config FWDTEN = OFF`. If its set to ON, you will need to change it to OFF.

If you end up modifying main.c, we recommend that you make main.c part of the VectorCAST project so that it's checked in and out of source control with the project. This removes the need for each developer or tester to modify the file each time VectorCAST is installed. Please email support@vector.com and instructions for adding main.c to your VectorCAST project can be provided.

2.6 Known mdb issues

The Microchip command line debugger, mdb, is java based and unfortunately can be very slow at times. The default testcase timeout is 30 seconds and is set in the DATA/microchip/mplabx/mplabx_execute.bat file in the VectorCAST install. If you are getting testcase timeouts try changing the timeout to 150 seconds and rerunning your testcase after an environment rebuild.

```
if "%1" == "UARTSIM" goto :UARTSIM
if "%1" == "SIM" goto :UARTSIM

type tgt_command.txt >> mdbcommand.txt
%VECTORCAST_DIR%\python.exe execute_microchip_mdb.py --debugger %1 --timeout 30 --log vcast_pic.elf
goto :END

:UARTSIM
type sim_command.txt >> mdbcommand.txt
call mdb.bat mdbcommand.txt
goto :END
```

Figure 5 – execute script timeout

There have also been cases where MPLAB X adds support for new processors and new debuggers, but support is not added for those processors and debuggers to mdb. If you get error messages indicating an unknown processor or an unknown debugger contact support@vector.com and we can help debug the issue.

2.7 Use Case: Debugging

Debugging VectorCAST testcases is done using MPLAB X. Once you right click on your testcase and select 'Execute with Debug', MPLAB X will open. Note that the path to the MPLAB X executable needs to be added to the path in your setup_vectorcast.bat script for VectorCAST to find MPLAB X.

To open the VectorCAST harness elf file go to File->Import->Hex/ELF ... File. Browse to vcast_pic.elf in your environment directory. Also choose your Device and your Hardware Tool. Then select Next and on the next screen click Finish.

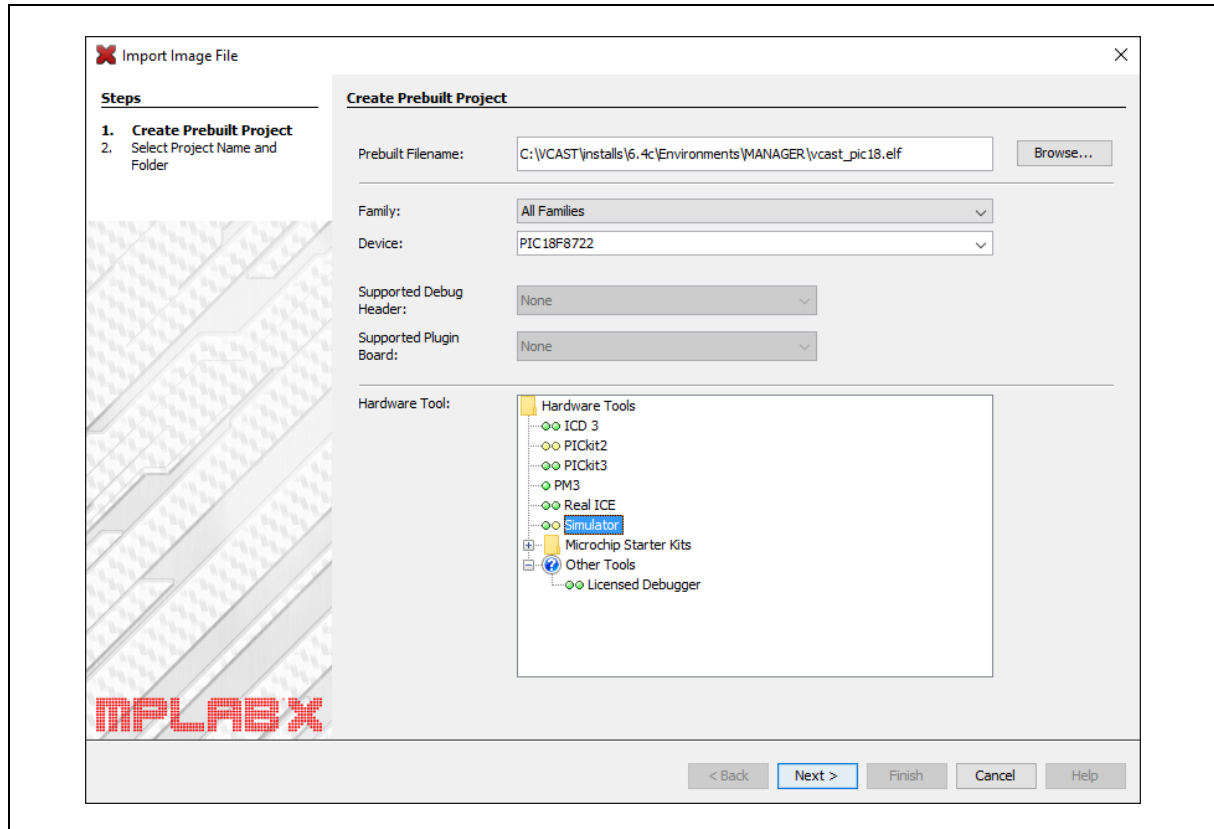


Figure 6 – MPLAB X

Once the project has opened click on Projects in the upper left-hand corner and under vcast_pic should be a folder called Source Files. Expand Source Files and find main.c and double click on it. Inside main() is a call to vcast_main(). Put a breakpoint on this line.

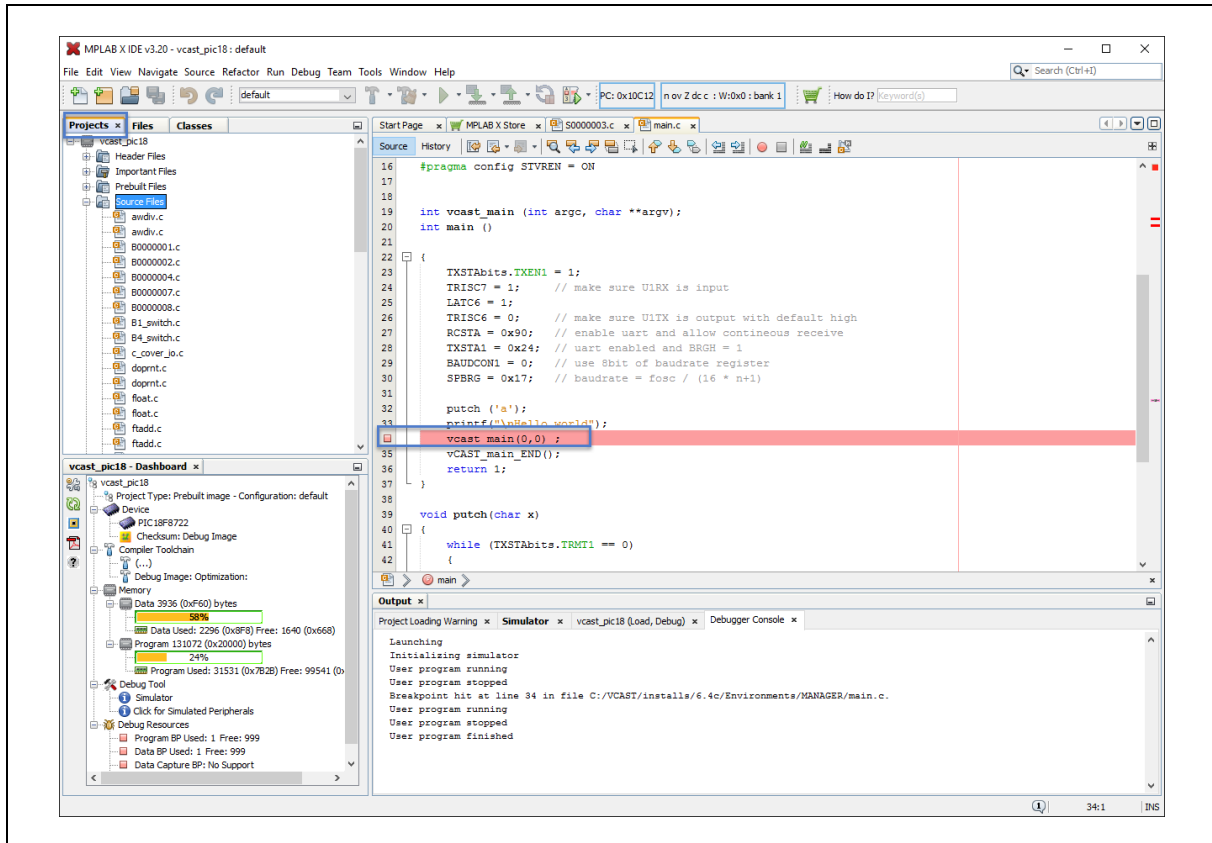


Figure 7 – Execute with Debug

To start the program running go to Debug->Debug Project (vcast_pic).

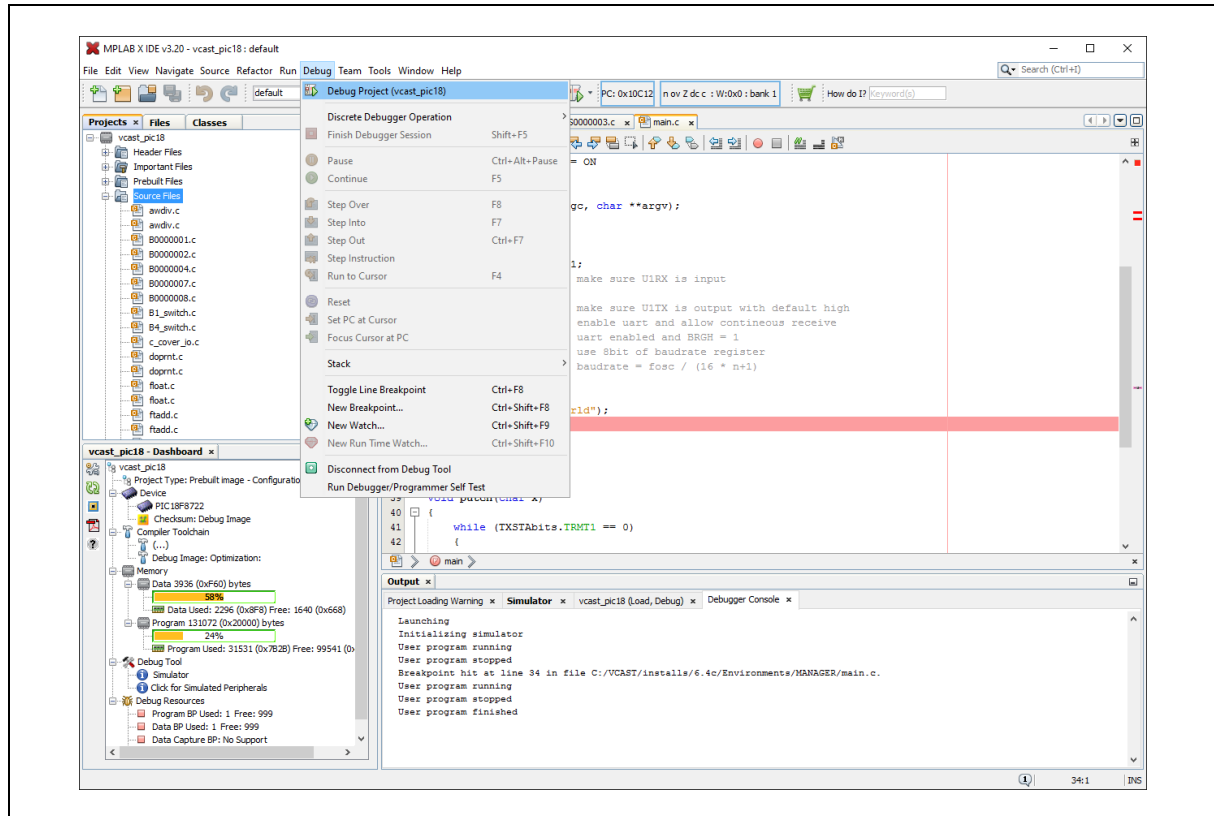


Figure 8 – Execute with Debug

If you want to set a breakpoint on your code, you can go to **Navigate->Go To Symbol** and type in the name of your function and it will take you to the source line where that symbol is.

3 Additional Resources

<https://www.microchip.com/>

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/>.