	<h1>SSE-310 MPS3</h1> <h2>BSP Pack User Guide</h2>
Version:	1.0.0
Date of Issue:	07/06/2022

© Copyright ARM Limited 2022. All rights reserved. Non-confidential.

## Contents

Contents.....	1
Introduction .....	2
Prerequisites.....	2
Documents .....	2
Pack Installation – Keil MDK.....	3
Import and build the example Vio project – Keil MDK .....	3
Running example on AWS with AVH image.....	4

## Introduction

This document is a general guide to use the SSE-310 MPS3 BSP pack. The CMSIS pack is to be used with the Corstone-SSE-310 AVH FVP. The pack contains necessary source files, a linker script file, and a specification document to kick start development for the Corstone-SSE-310 AVH FVP, a reference secure-side Blinky example to enable a user to understand uVision project configuration, and a reference Vio example to demonstrate the AVH capabilities for I/O simulation with Python scripts.

This document specifies system prerequisites and explains how to build and run the reference Vio example on the Corstone-SSE-310 AVH FVP. Blinky example can be build and run similarly.

## Prerequisites

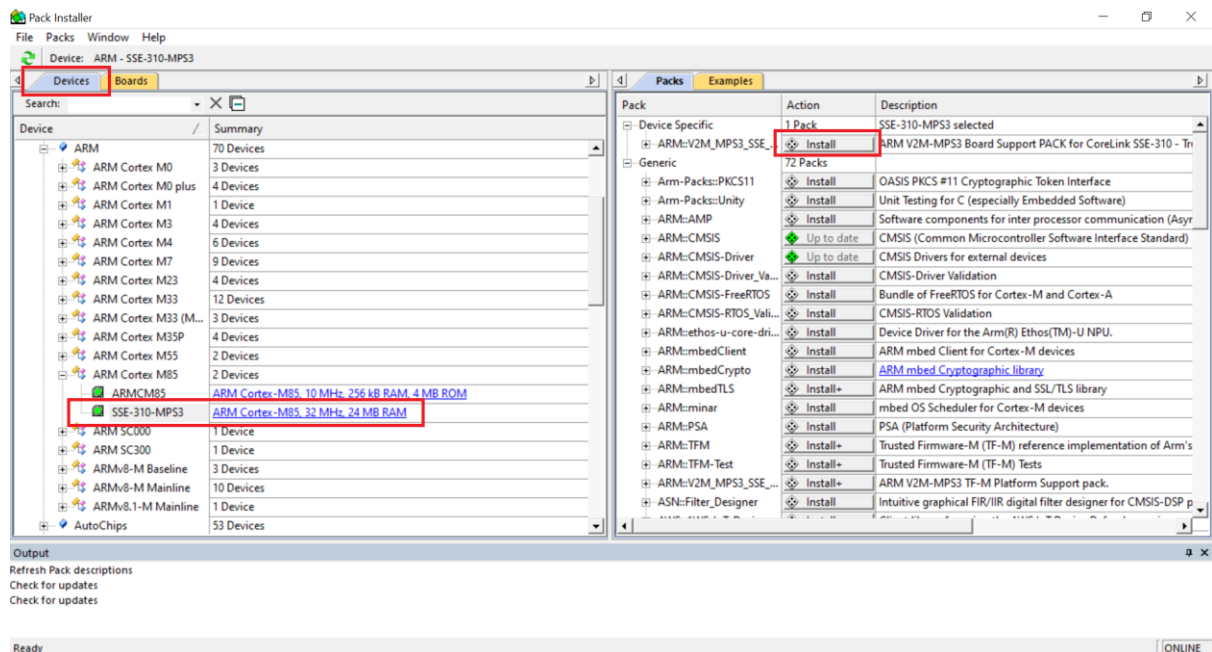
- Minimum [Keil MDK v5.37](#)

## Documents

1. Corstone-310 FVP Technical Overview: contains overview of the FVP and its features.
2. Arm Corstone SSE-310 Subsystem Technical Reference Manual: contains the specification of the architecture of the subsystem, description of several interfaces (address, data width, clock/power/reset domain), functional description of the components.

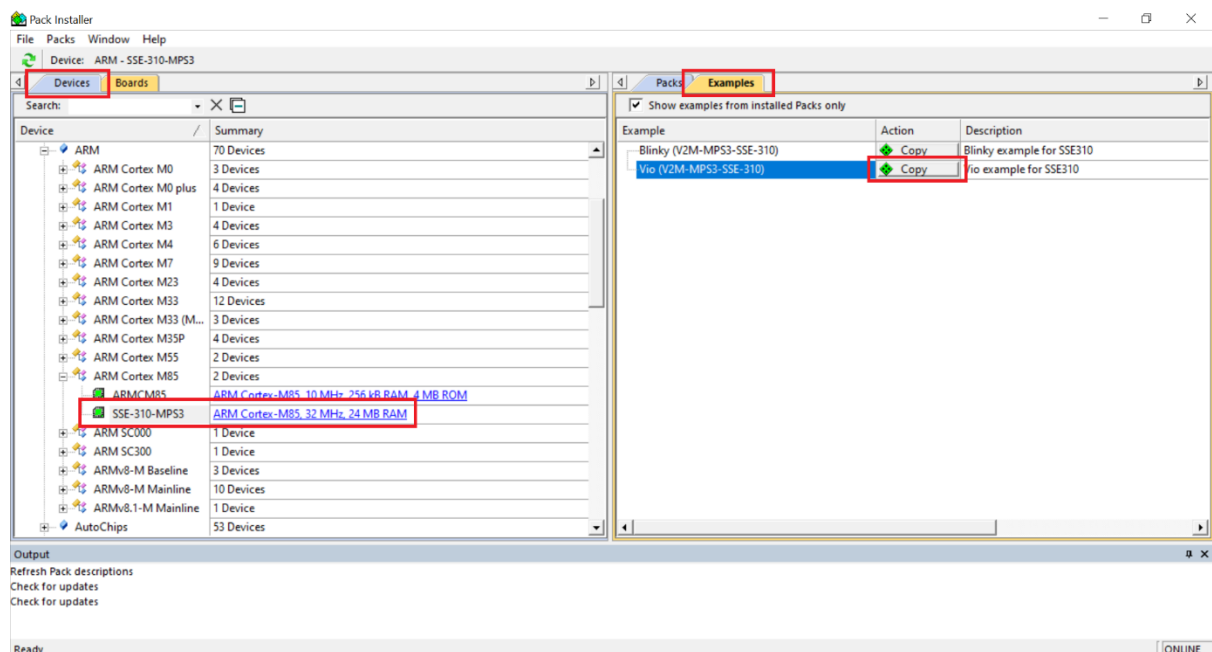
## Pack Installation – Keil MDK

Install ARM::V2M\_MPS3\_SSE\_310\_BSP using the Pack Installer. The pack can be browsed by selecting SSE-310-MPS3 device under ARM Cortex M85 Devices.



## Import and build the example Vio project – Keil MDK

Copy the Vio project using the Pack Installer. The example project can be found by searching and selecting SSE-310-MPS3 device under ARM Cortex M85 Devices, then selecting Examples tab on the right.



Once copied, open the Vio project using the uVision and simply build the Target1 listed inside Project Explorer.

Note: compile with C99 option (-std=c99)

## Running example on AWS with AVH image

To utilize the [Arm Virtual Hardware \(AVH\)](#), you will need to create an [AWS Account](#) if you don't already have one.

Launching the instance in EC2 ([AWS on getting started](#))

1. Go to [EC2](#) in the AWS Web Console.
2. Select **Launch Instances** which will take you to a wizard for launching the instance.
3. Choose an **Amazon Machine Image (AMI)** In the Search box, type *Arm Virtual Hardware* then find the item called "Arm Virtual Hardware" that is by Arm, and press **Select** for that item. This will raise a subscription page/pop-up titled, **Arm Virtual Hardware**. You will note that the subscription is free from Arm, but AWS does charge for the costs of the instances themselves according to the pricing chart provided. You must select **Continue** if you want to move forward.
4. **Choose an Instance Type** - Select one of the instance types from the list. Keep in mind that there are charges that accrue while the instance is running. From here you may select **Review and Launch** to move directly to the launch page or select **Next: Configure Instance Details** if you need to set any custom settings for this instance.

Once you complete the wizard by initiating the instance launch you will see a page that allows you to navigate directly to the new instance. You may click this link or go back to your list of instances and find the instance through that method.

Whichever way you choose find your new instance and select its instance ID to open the page to manage the instance.

Connecting to the instance:

1. Select **Connect** to open an SSH terminal session to the instance in your browser.
2. Ensure the User name field is set to *ubuntu*.
3. Select the **Connect** button to open the session. This will put you in a browser window where you will have an SSH terminal window ready for your input.

The Objects\Vio.axf can be copied to the instance with scp to "/home/ubuntu". To run the example:

```
VHT_Corstone_SSE-310 -C mps3_board.visualisation.disable-  
visualisation=1 -C mps3_board.telnetterminal0.start_telnet=0 -C  
mps3_board.uart0.out_file="-" -C  
mps3_board.uart0.unbuffered_output=1 --stat -C  
mps3_board.DISABLE_GATING=1 Vio.axf
```

To run python I/O simulation, download vio python example:

```
ubuntu@ip-172-31-30-219:~$ wget https://github.com/ARM-  
software/AVH/raw/main/interface/python/arm_vio.py
```

Edit the file and set verbosity level to debug in line 17-18.



```
VHT_Corstone_SSE-310 -V . -C mps3_board.visualisation.disable-  
visualisation=1 -C mps3_board.telnetterminal0.start_telnet=0 -C  
mps3_board.uart0.out_file="-" -C  
mps3_board.uart0.unbuffered_output=1 --stat -C  
mps3_board.DISABLE GATING=1 Vio.axf
```

```
ubuntu@ip-172-31-30-219:~$ VHT_Corstone_SSE310 -V . -C mps3_board.visualisation.disable-visualisation=1 -C mps3_board.telnetterminal0.start_telnet=0 -C mps3_board.uart0.out
last -C mps3_board.uart0.unbuffered_output=1 --stat -C mps3_board.DISABLE_GATTING=1 Vio.axf
Py: VIO: [INFO] Verbosity level is set to DEBUG
Py: VIO: [INFO] Python function init() called
telnetterminal5: Listening for serial connection on port 5000
telnetterminal2: Listening for serial connection on port 5001
telnetterminal1: Listening for serial connection on port 5002
telnetterminal0: Listening for serial connection on port 5003

Ethos-U rev 136b7d75 --- Nov 25 2021 12:11:30
(C) COPYRIGHT 2019-2021 Arm Limited
ALL RIGHTS RESERVED

Warning: cpu_core: mps3_board.sse300.timer0: CNTBase0 access(write) error: register 0x10 (CNTFRQ) is read-only
In file: (unknown):0
In process: cpu_core.thread_p_6 @ 0 s
Py: VIO: [INFO] Python function wrValue() called
Py: VIO: [DEBUG] Write value at index 0: 0
Py: VIO: [INFO] Python function rdValue() called
Py: VIO: [DEBUG] Read value at index 0: 0
Vio value is set to 0
Py: VIO: [INFO] Python function wrValue() called
Py: VIO: [DEBUG] Write value at index 0: 1
Py: VIO: [INFO] Python function rdValue() called
Py: VIO: [DEBUG] Read value at index 0: 1
Vio value is set to 1
Py: VIO: [INFO] Python function wrValue() called
Py: VIO: [DEBUG] Write value at index 0: 2
Py: VIO: [INFO] Python function rdValue() called
Py: VIO: [DEBUG] Read value at index 0: 2
Vio value is set to 2
Py: VIO: [INFO] Python function wrValue() called
Py: VIO: [DEBUG] Write value at index 0: 3
Py: VIO: [INFO] Python function rdValue() called
Py: VIO: [DEBUG] Read value at index 0: 3
Vio value is set to 3
Py: VIO: [INFO] Python function wrValue() called
Py: VIO: [DEBUG] Write value at index 0: 4
Py: VIO: [INFO] Python function rdValue() called
Py: VIO: [DEBUG] Read value at index 0: 4
Vio value is set to 4
```