

	<b>SSE-300 MPS3</b> BSP Pack User Guide
Version:	1.4.0
Date of Issue:	October 13, 2023

© Arm® Limited 2022-2023. All rights reserved. Non-confidential.

## Contents

<b>1 Concept</b>	<b>2</b>
1.1 Introduction	2
1.2 Prerequisites	2
1.3 Documents	3
<b>2 Installing the Pack</b>	<b>4</b>
2.1 Keil MDK	4
2.2 IAR Embedded Workbench	4
<b>3 Blinky example: Keil MDK</b>	<b>5</b>
3.1 Import	5
3.2 Build	5
3.2.1 Building with ARMCLANG	5
3.2.2 Building with GCC	5
3.3 Run and Debug	6
3.4 Run - Terminal	7
3.5 Run - FPGA	7
<b>4 Blinky example: IAR Embedded Workbench</b>	<b>10</b>
4.1 Import	10
4.2 Build	10
4.3 Run and Debug	11
4.4 Run - Terminal	11
4.5 Run - FPGA	12
<b>5 Vio example: Keil MDK</b>	<b>13</b>
5.1 Import	13
5.2 Build	13
5.2.1 Building with ARMCLANG	13
5.2.2 Building with GCC	13
5.3 Run and Debug	14
5.4 Run - Terminal	16
<b>6 Vio example: IAR Embedded Workbench</b>	<b>17</b>
6.1 Import	17
6.2 Build	17
6.3 Run and Debug	18
6.4 Run - Terminal	19
<b>7 Attachments</b>	<b>20</b>
7.1 Blinky example tree	20
7.2 Vio example tree	20

# 1 Concept

## 1.1 Introduction

This document is a general guide to use the SSE-300 MPS3 BSP pack. The CMSIS pack is to be used with the Corstone®-300 platform MPS3 FVP model or AN552 FPGA (AN552: Arm Corstone™ SSE-300 with Cortex®-M55 and Ethos™-U55: Example Subsystem for MPS3). The pack contains necessary source files, a linker script file, and a specification document to kick-start development for the Corstone-300 MPS3 platform, and a reference secure side Blinky example to enable a user to understand uVision and IAR project configuration. The pack also provides a System View Description (SVD) file for the platform to be used with the uVision and IAR debugger. This document specifies system prerequisites and explains how to build and run the reference Blinky example on the SSE-300 MPS3 FVP model and on the AN552 FPGA.

### Terms and Abbreviations:

Terms	Meaning
BSP	Board Support Pack
FVP	Fixed Virtual Platform
FPGA	Field Programmable Gate Array
VIO	Virtual I/O

## 1.2 Prerequisites

- **Development IDEs:**  
Minimum Keil MDK v5.38a or IAR Embedded Workbench v9.40,
- **GCC GNU Compiler:**  
GNU Arm Embedded Toolchain Arm GCC (Version 11.3 or newer),
- **FVP model:**  
Corstone SSE-300 MPS3 FVP model (tested on Version 11.22.35),
- **MPS3 FPGA:**  
AN552: Arm Corstone™ SSE-300 with Cortex®-M55 and Ethos™-U55 Example Subsystem for MPS3 FPGA (tested on Version 3.0).
- **Python version 3.9 for VIO examples**



Download **Python3.9** like the following on **Ubuntu** if the version at **apt** is not working:

```
$ sudo apt-get update && sudo apt-get upgrade
$ sudo apt-get install -y make build-essential libssl-dev zlib1g-dev libbz2-dev
libreadline-dev libsqlite3-dev wget curl llvm libncurses5-dev libncursesw5-dev xz-
utils tk-dev liblzma-dev tk-dev

$ wget https://www.python.org/ftp/python/3.9.18/Python-3.9.18.tgz
$ tar xzf Python-3.9.18.tgz
$ cd Python-3.9.18

$ ./configure --enable-optimizations
$ make -j "$(nproc)"
$ sudo make altinstall
```

## 1.3 Documents

1. Application Note AN552: AN552 is a Single Cortex-M55 FPGA implementation of the Arm Corstone SSE-300 with Cortex-M55 and Ethos™-U55 Example Subsystem that uses SIE-300, SIE-200 components. The application note states which components are included, the memory map of AN552, details about the FPGA peripherals.
2. Arm Corstone-300 Foundation IP Technical Overview: contains overview of the IP and its features.
3. Arm Corstone SSE-300 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.



SSE-310 MPS3 BSP Pack contains additional documentation in the "Documents" folder.

---

## 2 Installing the Pack

### 2.1 Keil MDK

Install ARM::V2M\_MPS3\_SSE\_300\_BSP using the Pack Installer. (Find: Project > Manage > Pack Installer) The pack can be browsed by selecting SSE-300-MPS3 device under ARM Cortex M55 Devices.

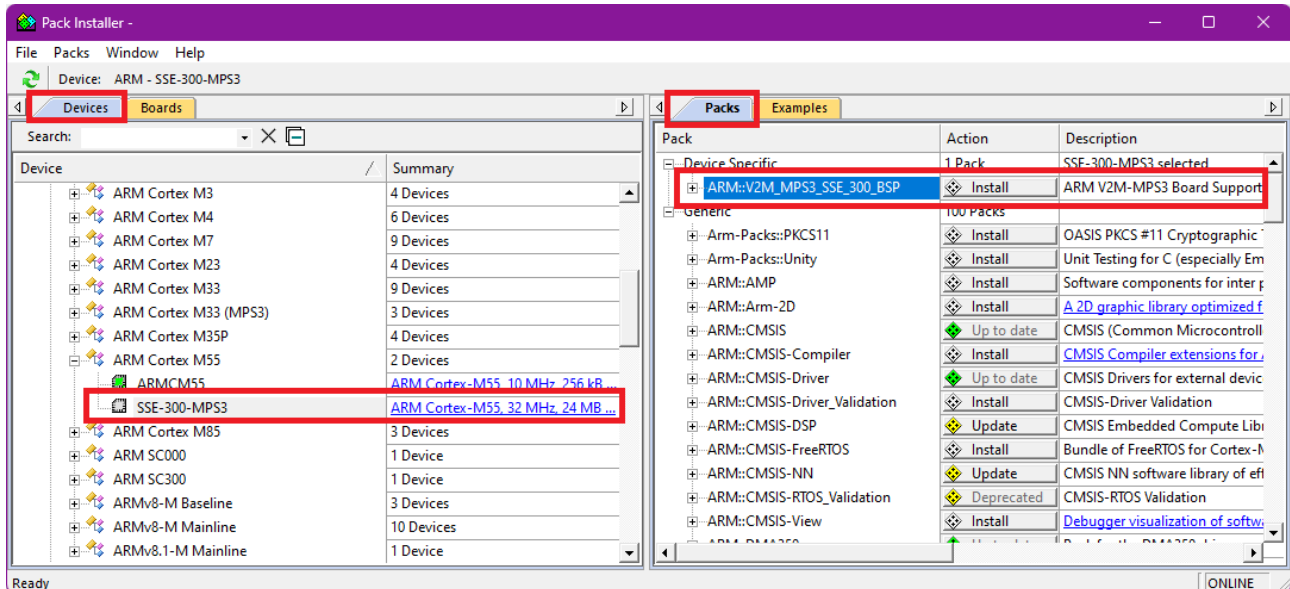


Figure 1: Selecting ARM::V2M\_MPS3\_SSE\_300\_BSP using uVision Pack Installer

### 2.2 IAR Embedded Workbench

Install ARM::V2M\_MPS3\_SSE\_300\_BSP using the CMSIS-Pack Manager. (Find: Project > CMSIS-Pack Manager) The pack can be browsed by selecting SSE-300-MPS3 device under ARM Cortex M55 Devices.

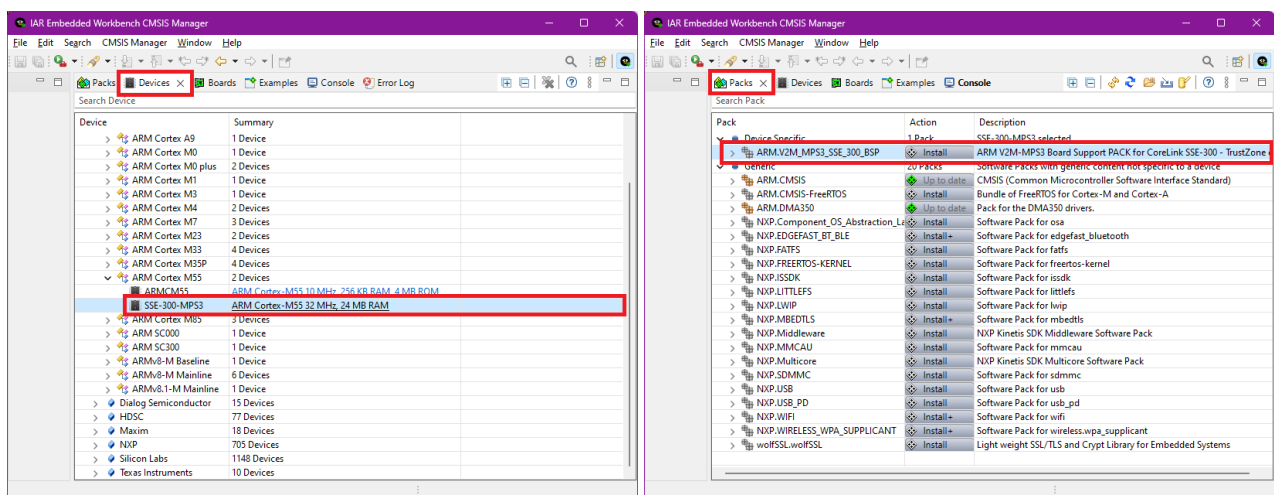


Figure 2: Selecting ARM::V2M\_MPS3\_SSE\_300\_BSP using IAR CMSIS-Pack Manager

## 3 Blinky example: Keil MDK

You can see the example's folder structure under [Blinky example tree](#) chapter.

### 3.1 Import

Copy the Blinky project using the Pack Installer. The example project can be found by searching and selecting V2M-MPS3-SSE-300-FVP Board under the Boards section.

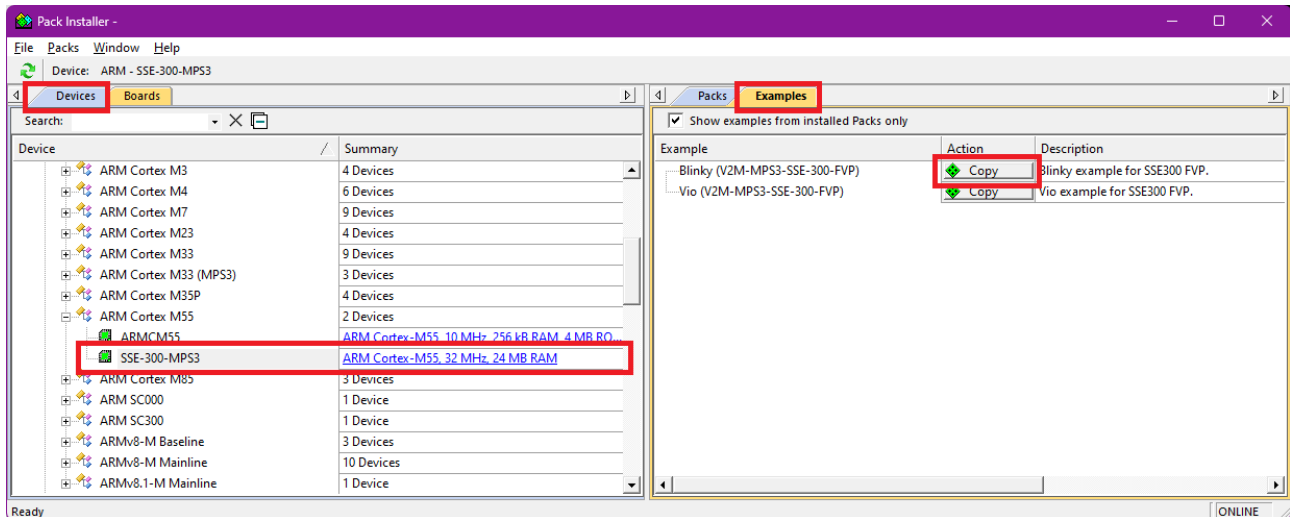


Figure 3: Keil MDK: Importing Blinky example

### 3.2 Build

#### 3.2.1 Building with ARMCLANG

Once copied, open the example if it is not already opened, then select target [ARMCLANG](#) and click on Build.

#### 3.2.2 Building with GCC

If you wish to compile the example using the [GNU Arm Embedded Toolchain Arm GCC](#) then select target [GCC](#) and click on Build. If the build fails, it is possible that you don't have the proper GNU compiler installed, or the specified installation folder does not match the one that you have. In this case please open the Manage Project Items window and set the GNU compiler to its proper installation directory. (Find: [GCC \(Target\)](#) > Manage Project Items)



Supported toolchain versions are GNU Arm Embedded Toolchain Arm GCC version 11.3 or newer.

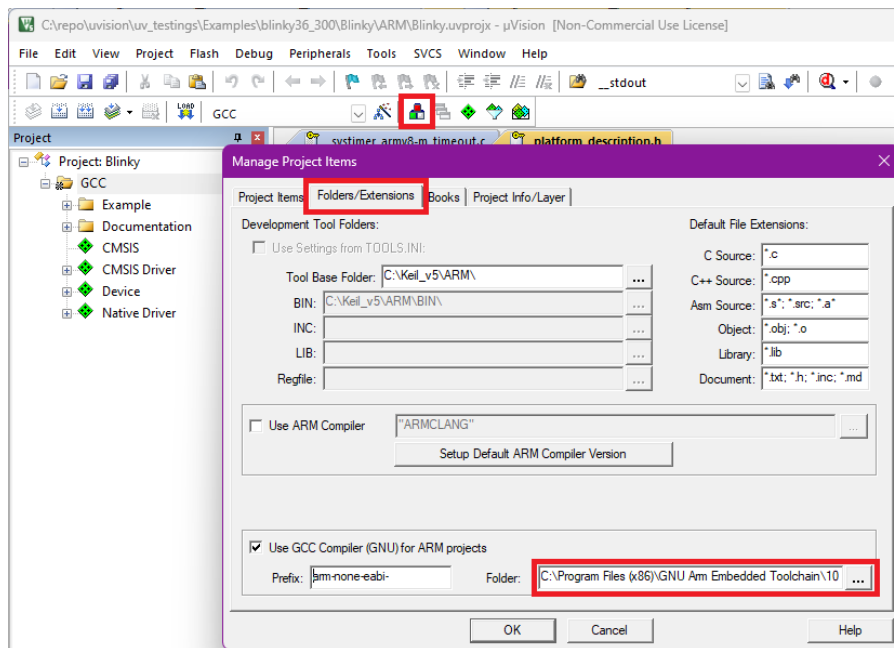


Figure 4: Keil MDK: Selecting GNU compiler

### 3.3 Run and Debug

This section explains how to run the Blinky example on the Corstone SSE-300 FVP model. First, download and install the SSE-300 FVP from the link provided in the [Prerequisites](#) section.

To run and debug the example using the FVP inside the uVision software, follow the steps below.

1. In the project section right click on the ARMCLANG/GCC target and click on "Options for Target".

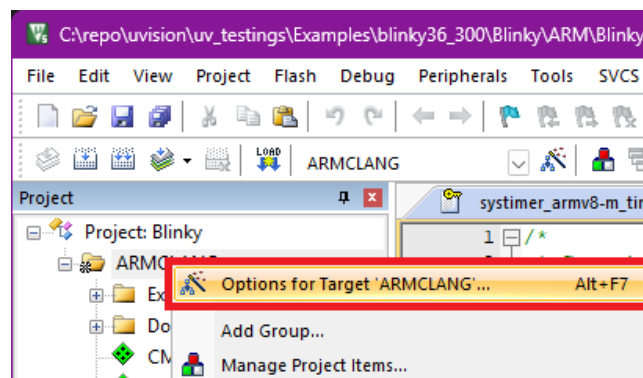


Figure 5: Keil MDK: Open target options

2. Click the Debug tab to open the debug settings. Then in the drop-down selection for the debugger, select "Models ARMv8-M Debugger", then click the Settings button next to it.

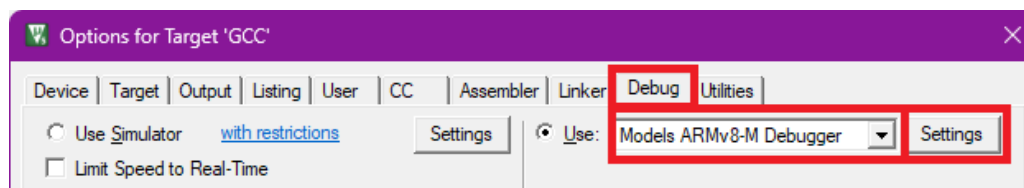


Figure 6: Keil MDK: Changing project Debugger

3. Open the "Models ARMv8-M Target Driver Setup" box and browse to the FVP\_Corstone\_SSE-300\_Ethos-U55 executable which is present inside the installation directory. Click OK and save the changes.

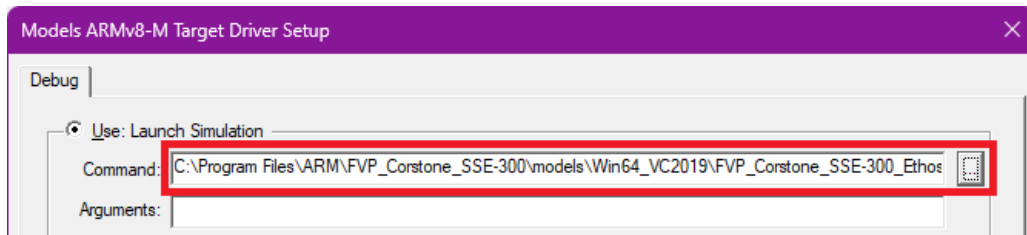


Figure 7: Keil MDK: Setting FVP

4. Build the target if it is not built, and then click the Debug button at the top to start a Debug session.

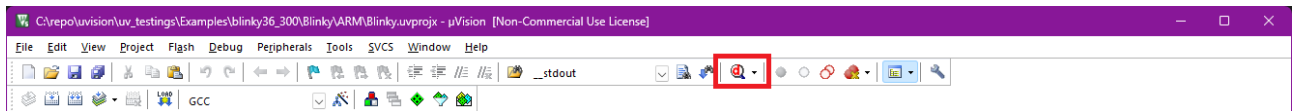


Figure 8: Keil MDK: Running and Debugging the example

The FVP window should pop up with code stopped at the entry breakpoint. On starting code execution, the LEDs in the FVP display can be seen to blink cyclically. You can use the debugger to stop, step, and set breakpoints inside the code.

### 3.4 Run - Terminal

After building the target, you can launch the FVP from the terminal, using the command:

```
<path_to_fvp>/FVP_Corstone_SSE-300_Ethos-U55.exe -a <path_to_axf>/Blinky.axf
```

### 3.5 Run - FPGA

After building the Blinky example, locate the Blinky.axf in the project output folder. To create binary file for the FPGA, the fromelf utility can be used, which is by default, located in the Keil install directory (C:/Keil\_v5/ARM/ARMCLANG/bin/fromelf.exe).



If you have are using a revision **B** board, use the **HBI0309B** directory. The examples are only verified on the revision **C** boards.

#### For the Blinky example, use this command:

```
fromelf.exe -bincombined -output Blinky.bin Blinky.axf
```

Copy the binary to the FPGA's SD card (x:/SOFTWARE) and set the address in images.txt (x:/MB/HBI0309C/AN552/images.txt) to 0x00000000, then restart the FPGA.

```
IMAGE0ADDRESS: 0x00000000 ;  
IMAGE0UPDATE: AUTO ;  
IMAGE0FILE: /SOFTWARE/Blinky.bin ;
```

In the IDE, in the "Options for Target...", in the Debug tab, select "CMSIS-DAP ARMv8-M Debugger". Depending on your setup, you can also use "ULINK Pro ARMv8-M Debugger". Click Settings.

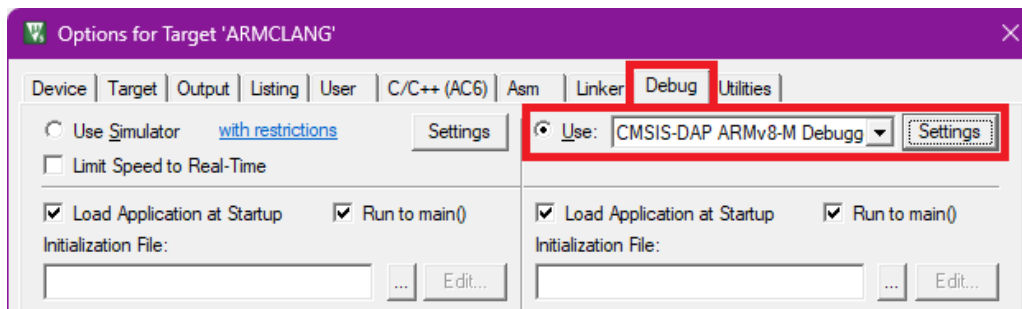


Figure 9: Keil MDK: Changing project Debugger

Make sure that on Debug tab, at Debug section, Reset after Connect is checked and Reset is set to Autodetect.

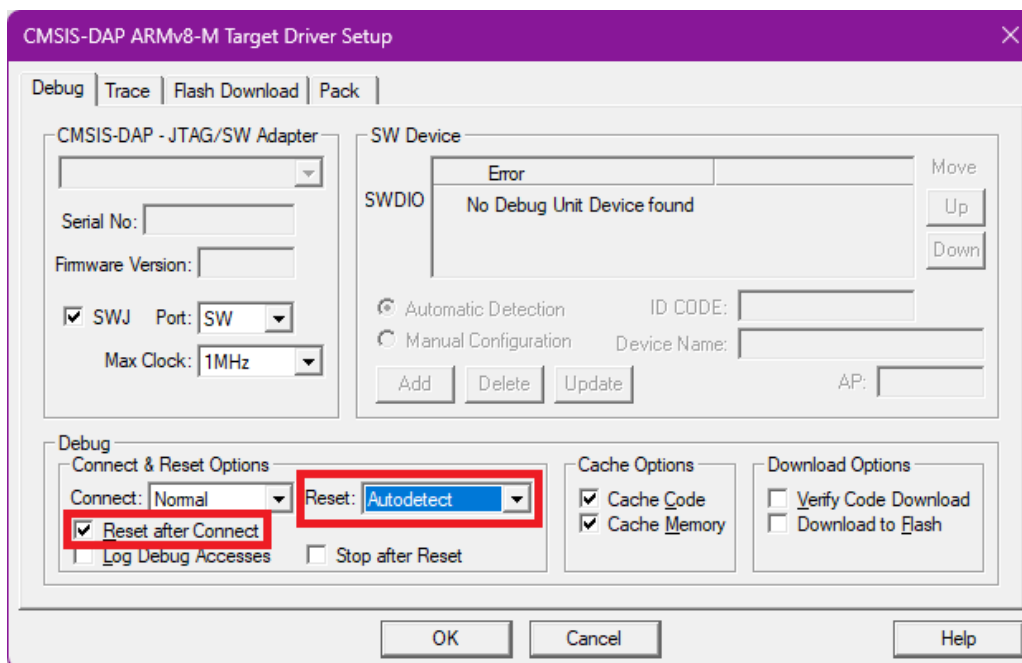


Figure 10: Keil MDK: "CMSIS-DAP ARMv8-M Debugger" settings

Verify that in the Flash Download tab, "Do not Erase" is selected and none of the checkboxes are checked. In the Pack tab, Enable is unchecked. Click OK.

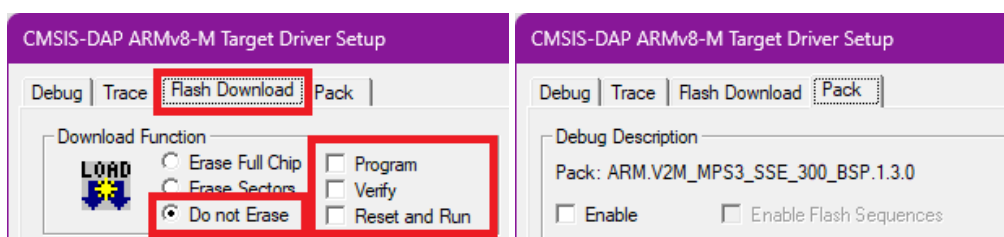


Figure 11: Keil MDK: "CMSIS-DAP ARMv8-M Debugger" settings - Flash Download and Pack

Then at "Utilities" tab, make sure that "Update Target" before Debugging is unchecked.



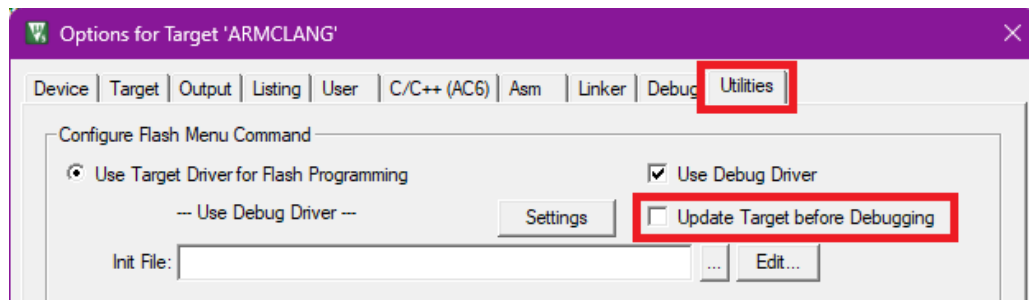


Figure 12: Keil MDK: Uncheck "Update Target" on "Utilities" tab in uVision

Click the debug button at top to start a debug session.

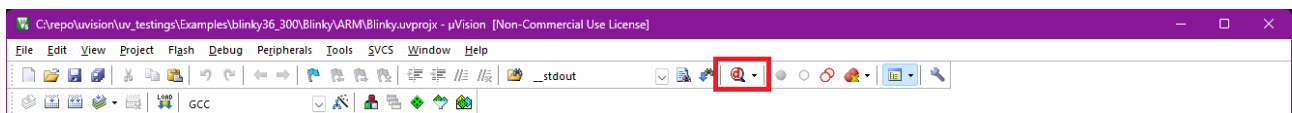


Figure 13: Keil MDK: Running and Debugging the example

After connection, reset is only possible when the target is running. If you wish to reset the target, make sure to run the target before pressing reset.

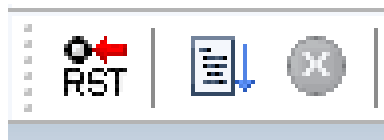


Figure 14: Keil MDK: Reset button

## 4 Blinky example: IAR Embedded Workbench

You can see the example's folder structure under [Blinky example tree](#) chapter.

### 4.1 Import

To import the project,

1. Create a new workspace, (File > New Workspace)
2. Open "CMSIS-Pack Manager" and chose a workspace folder, (Project > CMSIS-Pack Manager)
3. Select the SSE-300-MPS3 on the Device tab,
4. Import the "Blinky (V2M-MPS3-SSE-300-FVP)" example on the Examples tab.

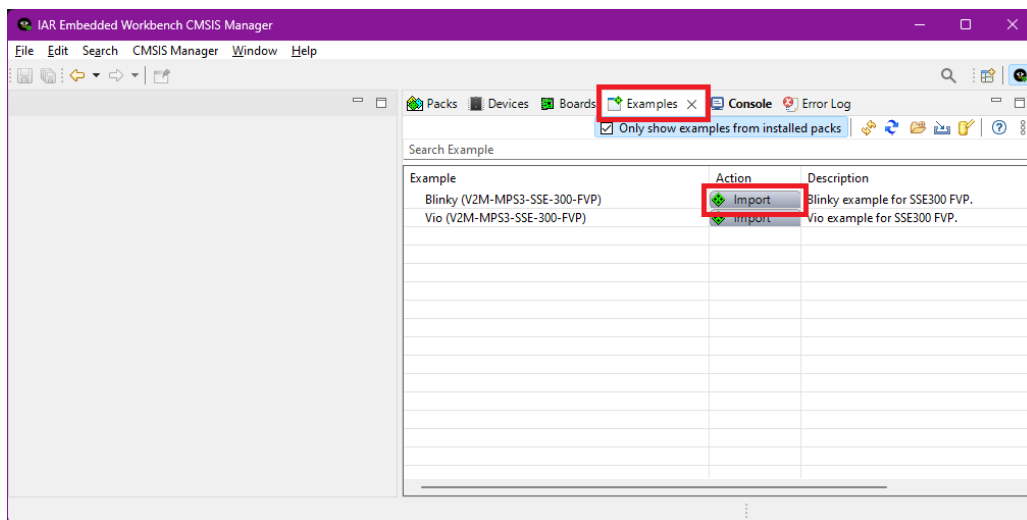


Figure 15: IAR EW: Importing the Blinky example



The example is only shown when the proper pack or menu is selected.

### 4.2 Build

To build the example click on the "Make" button, or press "F7".

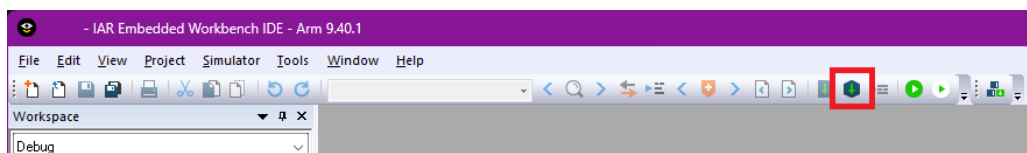


Figure 16: IAR EW: Building the Blinky example

## 4.3 Run and Debug

This section explains how to run the Blinky example on the Corstone SSE-300 FVP model. First, download and install the SSE-300 FVP from the link provided in the [Prerequisites](#) section.

To debug the example inside the IAR Embedded Workbench software, follow the steps below.

### Start up the FVP with a CADI server:

```
<path_to_fvp>/FVP_Corstone_SSE-300_Ethos-U55.exe -S
```

### Set up the debugger inside IAR:

Open the project Options, by clicking on the root file in the Workspace and then selecting it from the Project menu item. (Project > Options)

Select the Debugger in the Category list and select CADI as Driver on the Setup tab.

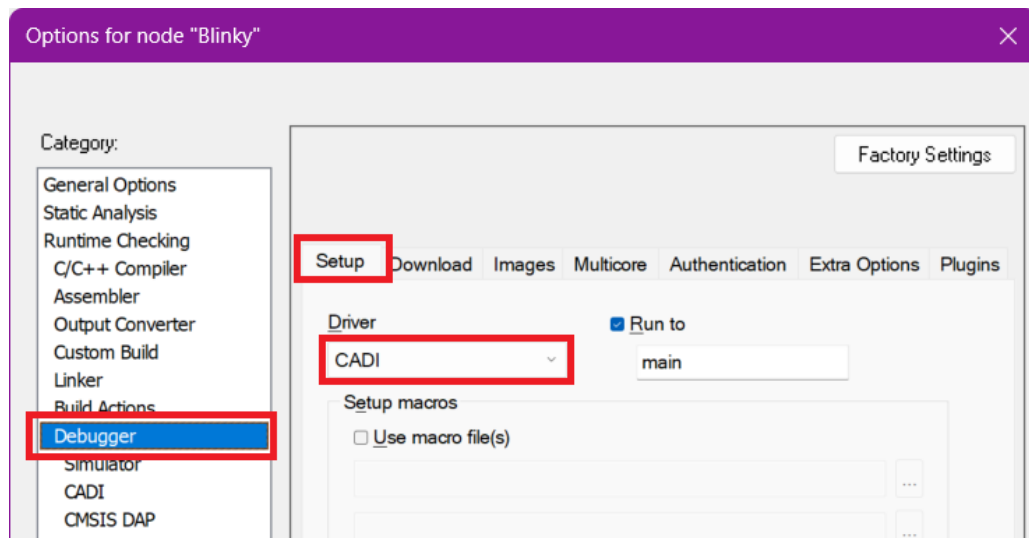


Figure 17: IAR EW: Selecting Debugger



You can select the desired CADI server in the CADI Category if there are more than one you wish to use. Otherwise, it can be left empty.

To run the debugger click on the "Download and Debug" button.

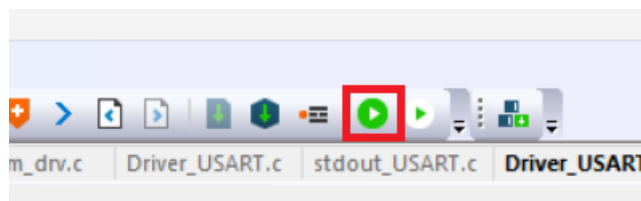


Figure 18: IAR EW: Running the example with CADI server

## 4.4 Run - Terminal

After building the target, you can launch the FVP from the terminal, using the command:

```
<path_to_fvp>/FVP_Corstone_SSE-300_Ethos-U55.exe <path_to_bin>/Blinky.out
```

## 4.5 Run - FPGA

Before building the Blinky example activate the Output Conversion. Open the project Options, by clicking on the root file in the Workspace and then selecting it from the Project menu item.

Select the "Output Converter" Category, check in the "Generate additional output" checkbox and select the "Raw binary" output format.

(Project > Options > Output Converter > Output format)

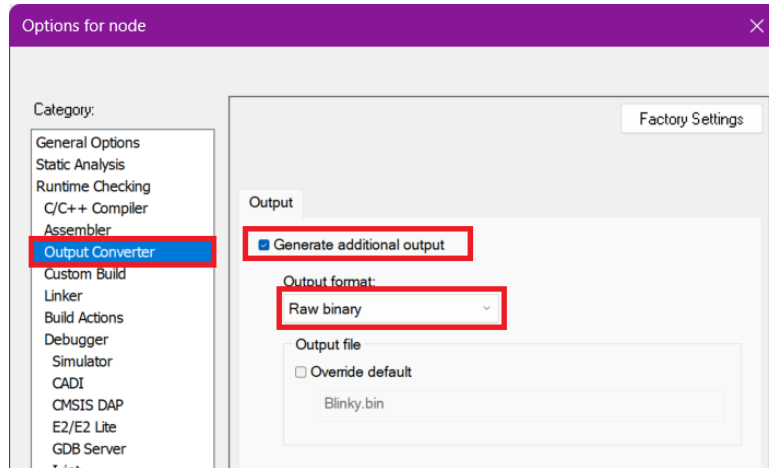


Figure 19: IAR EW: Building binary file

**Copy the binary** to the FPGA's SD card (x:/SOFTWARE) and set the address in images.txt (x:/MB/HBI0309C/AN552/images.txt) to 0x00000000, then restart the FPGA.

```
IMAGE0ADDRESS: 0x00000000 ;  
IMAGE0UPDATE: AUTO ;  
IMAGE0FILE: /SOFTWARE/Blinky.bin ;
```

**To debug** the project:

1. Select the "CMSIS DAP" Driver, and connect to the running example afterwards.  
(Project > Options > Debugger > Setup > Driver)
2. Click on "Run and Debug".



If you have are using a revision **B** board, use the **HBI0309B** directory. The examples are only verified on the revision **C** boards.

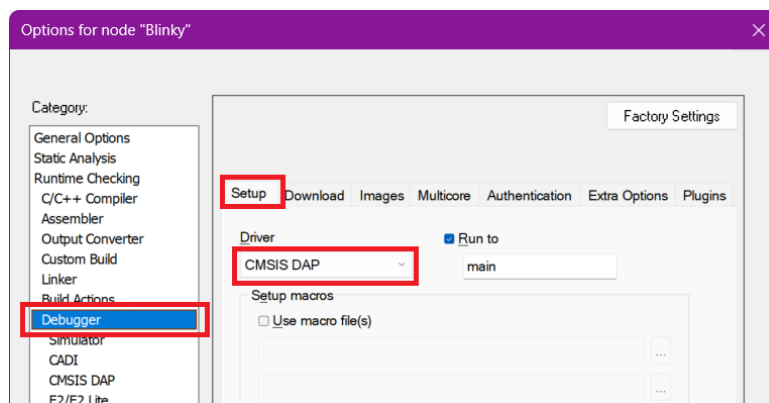


Figure 20: IAR EW: Select "CMSIS DAP" Debugger for MPS3 FPGA

## 5 Vio example: Keil MDK

You can see the example's folder structure under [Vio example tree](#) chapter.



To run this example you need FVP version 11.23 or higher!

### 5.1 Import

Copy the Vio project using the Pack Installer. The example project can be found by searching and selecting V2M-MPS3-SSE-300-FVP Board under the Boards section.

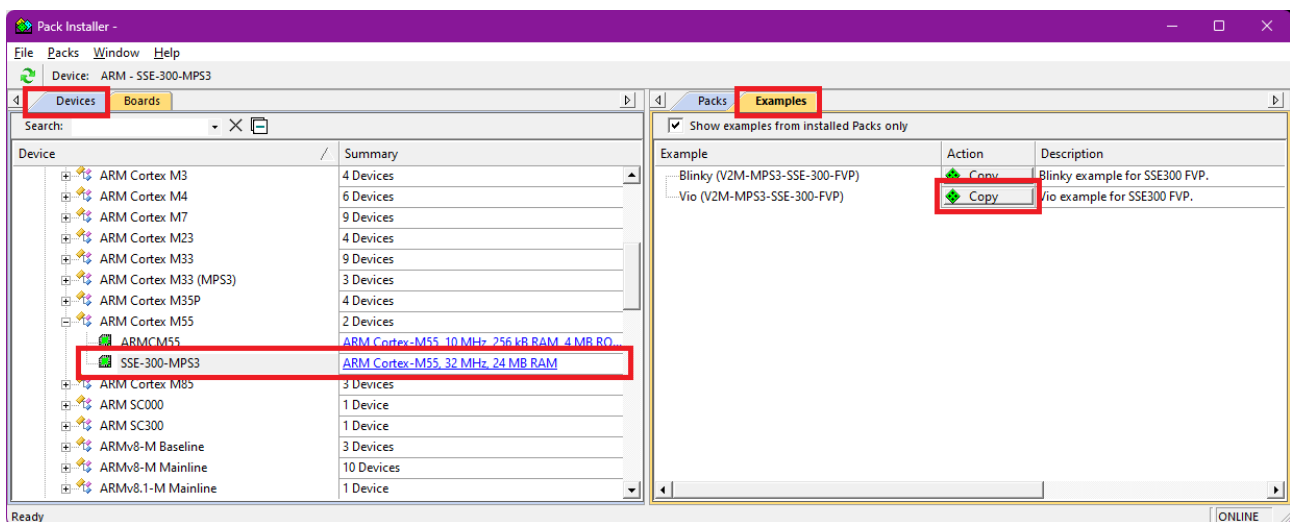


Figure 21: Keil MDK: Importing the Vio example

### 5.2 Build

#### 5.2.1 Building with ARMCLANG

Once copied, open the example if it is not already opened, then select target [ARMCLANG](#) and click on Build.

#### 5.2.2 Building with GCC

If you wish to compile the example using the [GNU Arm Embedded Toolchain Arm GCC](#) then select target [GCC](#) and click on Build. If the build fails, it is possible that you don't have the proper GNU compiler installed, or the specified installation folder does not match the one that you have. In this case please open the Manage Project Items window and set the GNU compiler to its proper installation directory. (Find: [GCC \(Target\)](#) > Manage Project Items)



Supported toolchain versions are GNU Arm Embedded Toolchain Arm GCC version 11.3 or newer.

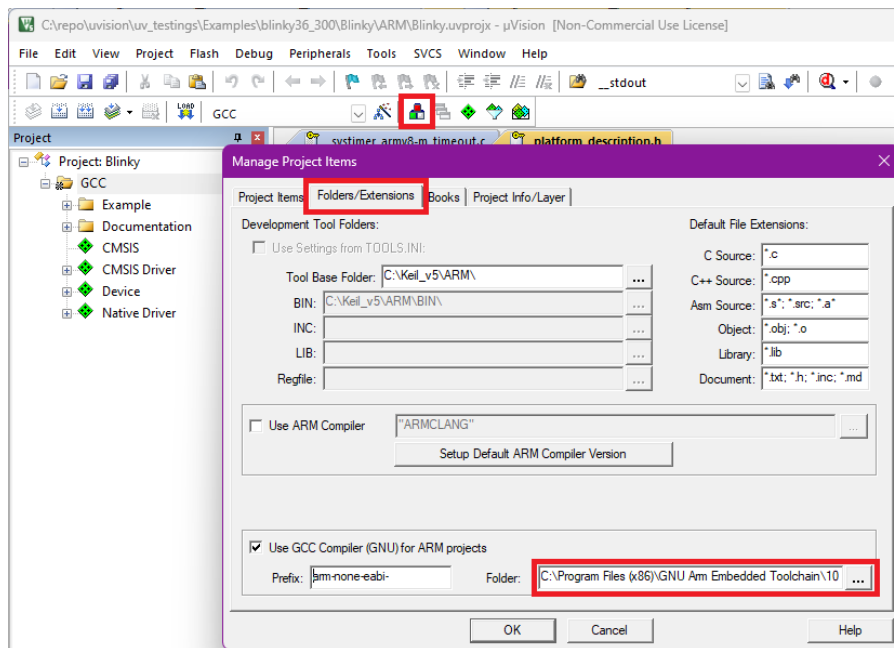


Figure 22: Keil MDK: Selecting GNU compiler

## 5.3 Run and Debug

This section explains how to run the Blinky example on the Corstone SSE-300 FVP model. First, download and install the SSE-300 FVP from the link provided in the [Prerequisites](#) section.



The VIO example might not work with newer Python versions. To run the example use version Python3.9.x.

To run and debug the example using the FVP inside the uVision software, follow the steps below.

1. Download [arm\\_vio.py](#), like the following:

PS > wget https://github.com/ARM-software/AVH/raw/main/interface/python/arm\_vio.py -OutFile arm\_vio.py  
Then uncomment the `#verbosity = logging.DEBUG` line in it.

```

15
16  ## Set verbosity level
17  verbosity = logging.DEBUG
18  #verbosity = logging.ERROR
19

```

Figure 23: Keil MDK: Setting debug mode in `arm_vio.py`

2. Set environmental variables:

### On Windows:

PS > \$env:PYTHONHOME = 'C:/Users/<user>/AppData/Local/Programs/Python/Python39'

### On Linux:

\$ export PYTHONPATH=/usr/local/lib/python3.9:/usr/local/lib/python3.9/lib-dynload

3. In the project section right click on the ARMCLANG/GCC target and click on "Options for Target".

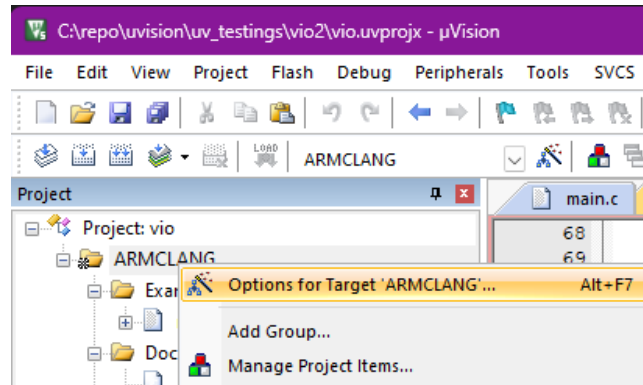


Figure 24: Keil MDK: Open target options

4. Click the Debug tab to open the debug settings. Then in the drop-down selection for the debugger, select "Models ARMv8-M Debugger", then click the Settings button next to it.

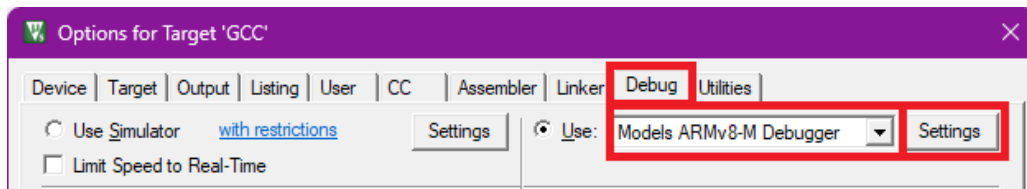


Figure 25: Keil MDK: Changing project Debugger

5. Open the "Models ARMv8-M Target Driver Setup" box and browse to the FVP\_Corstone\_SSE-300\_Ethos-U55 executable. Set the "Arguments" to the arm\_vio.py container folder. (Like: -C mps3\_board.v\_path=C:\repo\vio) Click OK and save the changes.

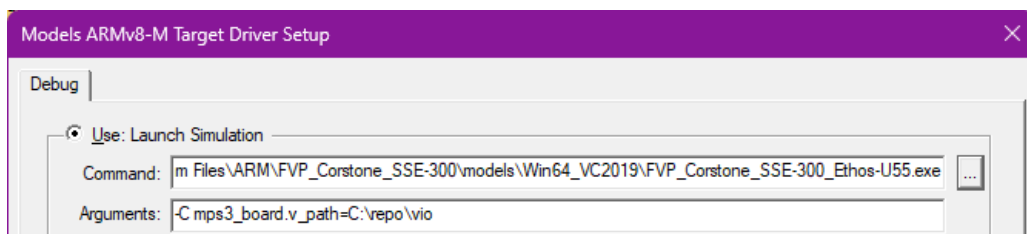


Figure 26: Keil MDK: Setting FVP

6. Build the target if it is not built, and then click the Debug button at the top to start a Debug session. The FVP window should pop up with code stopped at the entry breakpoint. You can use the debugger to stop, step, and set breakpoints inside the code.

**If the example is running**, you should see the numbers incrementing both on the serial, and inside the running arm\_vio.py script that you see in the MDK. (see: Keil MDK: Running Vio Example output)

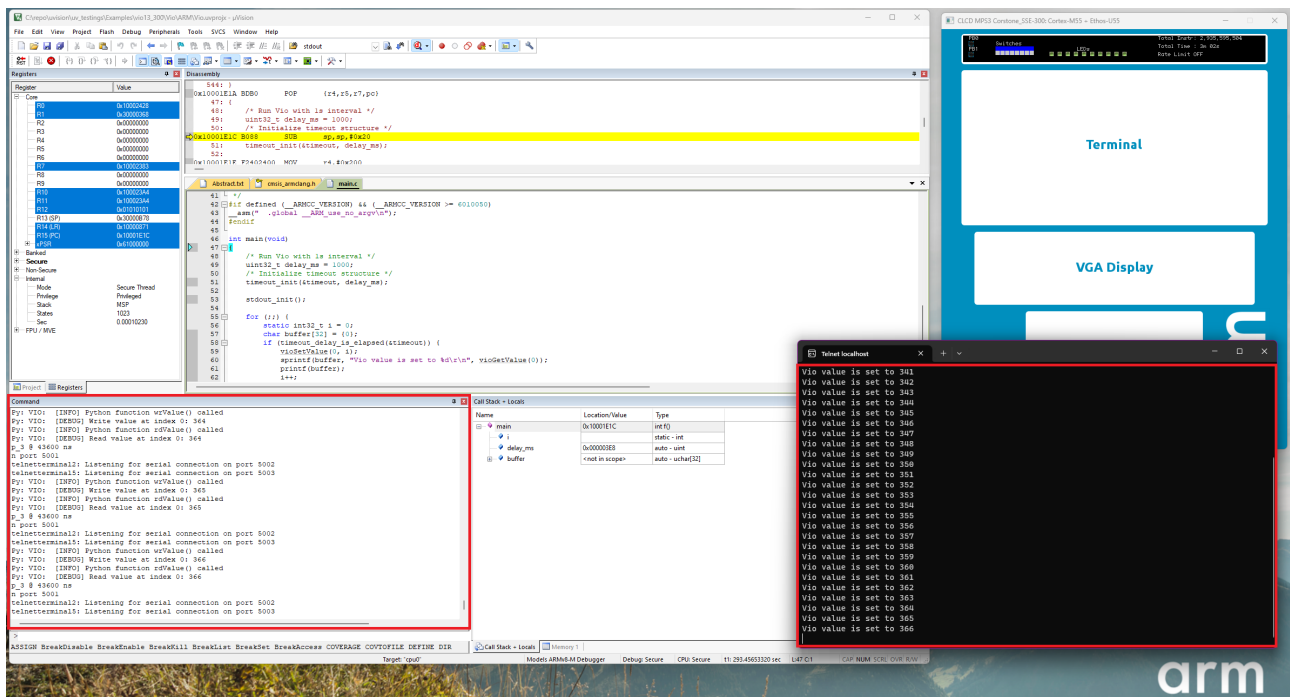


Figure 27: Keil MDK: Running Vio Example output

## 5.4 Run - Terminal

### 1. Download [arm\\_vio.py](https://github.com/ARM-software/AVH/raw/main/interface/python/arm_vio.py), like the following:

PS > wget https://github.com/ARM-software/AVH/raw/main/interface/python/arm\_vio.py -OutFile arm\_vio.py  
Then uncomment the #verbosity = logging.DEBUG line in it.

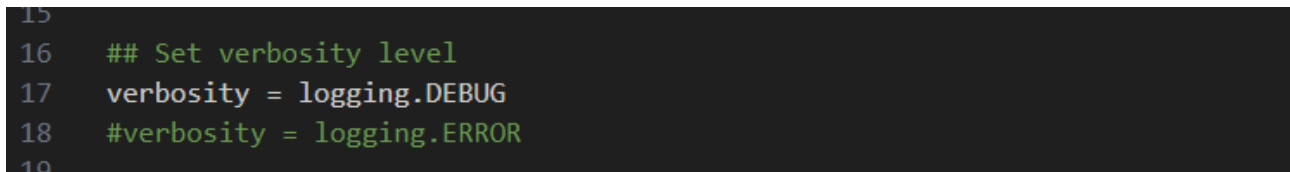


Figure 28: Keil MDK: Setting debug mode in arm\_vio.py

### 2. Set environmental variables to run the example:

#### On Windows:

PS > \$env:PYTHONHOME = 'C:/Users/<user>/AppData/Local/Programs/Python/Python39'

#### On Linux:

\$ export PYTHONPATH=/usr/local/lib/python3.9:/usr/local/lib/python3.9/lib-dynload

### 3. Execute the Example with the following command:

PS > <path\_to\_fvp>/FVP\_Corstone\_SSE-300\_Ethos-U55.exe -a <path\_to\_axf>/Vio.axf  
-C mps3\_board.v\_path=<path\_to\_arm\_vio.py\_folder>



The VIO example might not work with some newer Python versions. To run the example use version Python3.9.x.



## 6 Vio example: IAR Embedded Workbench

You can see the example's folder structure under [Vio example tree](#) chapter.

### 6.1 Import

To import the project,

1. Create a new workspace, (File > New Workspace)
2. Open "CMSIS-Pack Manager" and chose a workspace folder, (Project > CMSIS-Pack Manager)
3. Select the SSE-300-MPS3 on the Device tab,
4. Import the "Vio (V2M-MPS3-SSE-300-FVP)" example on the Examples tab.

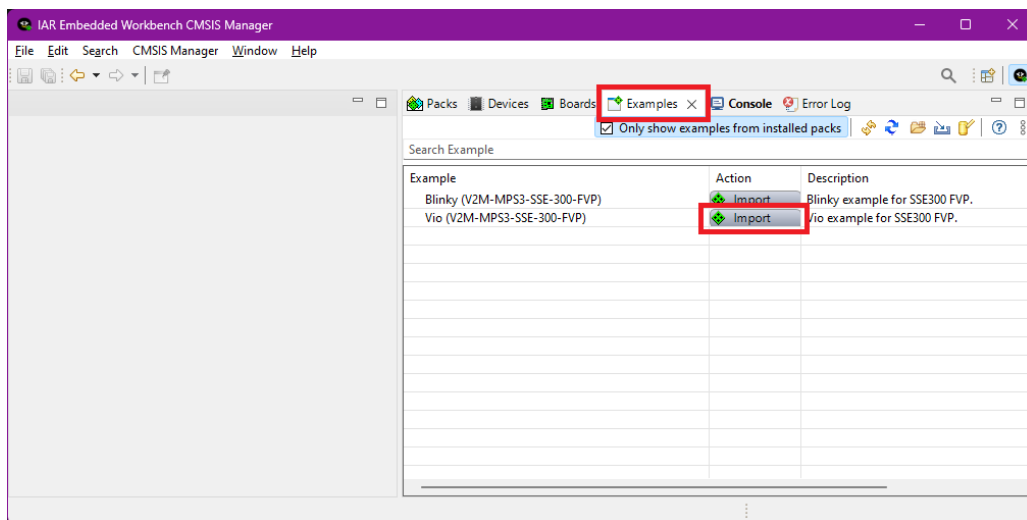


Figure 29: IAR EW: Importing the Vio example



The example is only shown when the proper pack or menu is selected.

### 6.2 Build

To build the example click on the "Make" button, or press "F7".

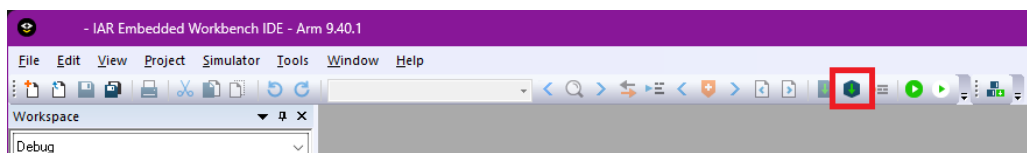


Figure 30: IAR EW: Building the Vio example



The VIO example might not work with newer Python versions. To run the example use version Python3.9.x.

## 6.3 Run and Debug

This section explains how to run the Vio example on the Corstone SSE-300 FVP model. First, download and install the SSE-300 FVP from the link provided in the [Prerequisites](#) section.

To debug the example inside the IAR Embedded Workbench software, follow the steps below.

Download [arm\\_vio.py](#), like the following:

```
PS > wget https://github.com/ARM-software/AVH/raw/main/interface/python/arm_vio.py -OutFile arm_vio.py
```

Then uncomment the `#verbosity = logging.DEBUG` line in it.

```
15
16  ## Set verbosity level
17  verbosity = logging.DEBUG
18  #verbosity = logging.ERROR
19
```

Figure 31: IAR EW: Setting debug mode in arm\_vio.py

### Set environmental variables:

#### On Windows:

```
PS > $env:PYTHONHOME = 'C:/Users/<user>/AppData/Local/Programs/Python/Python39'
```

#### On Linux:

```
$ export PYTHONPATH=/usr/local/lib/python3.9:/usr/local/lib/python3.9/lib-dynload
```

### Start up the FVP with a CADI server:

```
<path_to_fvp>/FVP_Corstone_SSE-300_Ethos-U55.exe -S -C mps3_board.v_path=<path_to_arm_vio.py_folder>
```

### Configure the debugger inside IAR:

Open the project Options, by clicking on the root file in the Workspace and then selecting it from the Project menu item. (Project > Options)

Select the Debugger in the Category list and select CADI as Driver on the Setup tab.

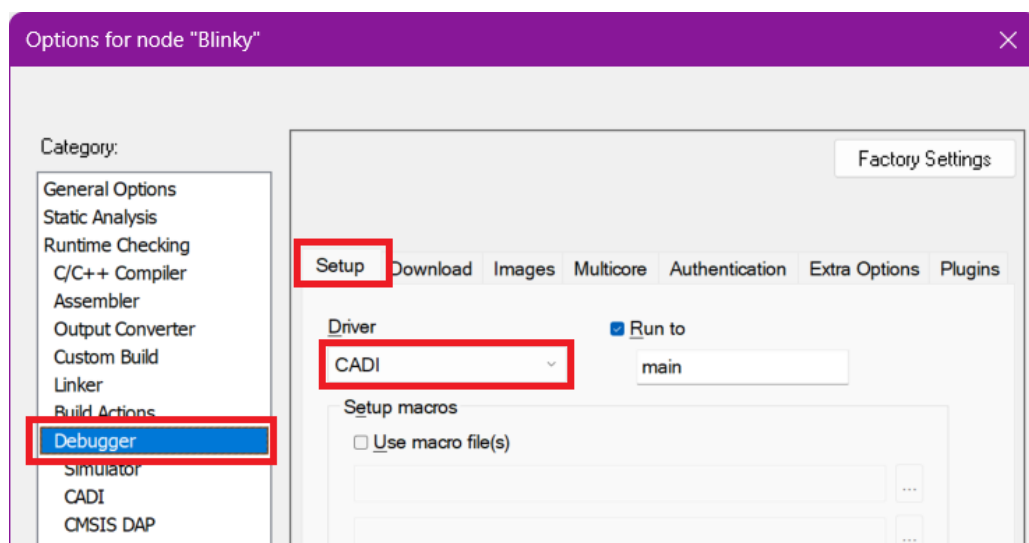


Figure 32: IAR EW: Selecting Debugger



You can select the desired CADI server in the CADI Category if there are more than one you wish to use. Otherwise, it can be left empty.

---

To run the debugger click on the "Download and Debug" button.

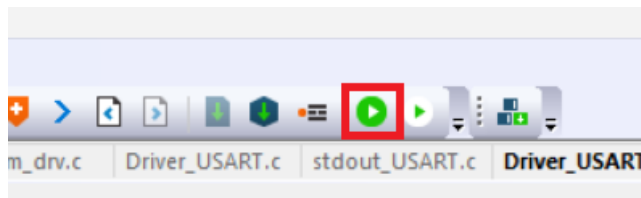


Figure 33: IAR EW: Running the example with CADI server

## 6.4 Run - Terminal

1. Download [arm\\_vio.py](#), like the following:

```
PS > wget https://github.com/ARM-software/AVH/raw/main/interface/python/arm_vio.py -OutFile arm_vio.py
```

Then uncomment the `#verbosity = logging.DEBUG` line in it.

```
15
16  ## Set verbosity level
17  verbosity = logging.DEBUG
18  #verbosity = logging.ERROR
19
```

Figure 34: IAR EW: Setting debug mode in `arm_vio.py`

2. Set environmental variables to run the example:

**On Windows:**

```
PS > $env:PYTHONHOME = 'C:/Users/<user>/AppData/Local/Programs/Python/Python39'
```

**On Linux:**

```
$ export PYTHONPATH=/usr/local/lib/python3.9:/usr/local/lib/python3.9/lib-dynload
```

3. Execute the Example with the following command:

```
PS > <path_to_fvp>/FVP_Corstone_SSE-300_Ethos-U55.exe <path_to_out>/Vio.out
-C mps3_board.v_path=<path_to_arm_vio.py_folder>
```



The VIO example might not work with newer Python versions. To run the example use version Python3.9.x.

---

## 7 Attachments

### 7.1 Blinky example tree

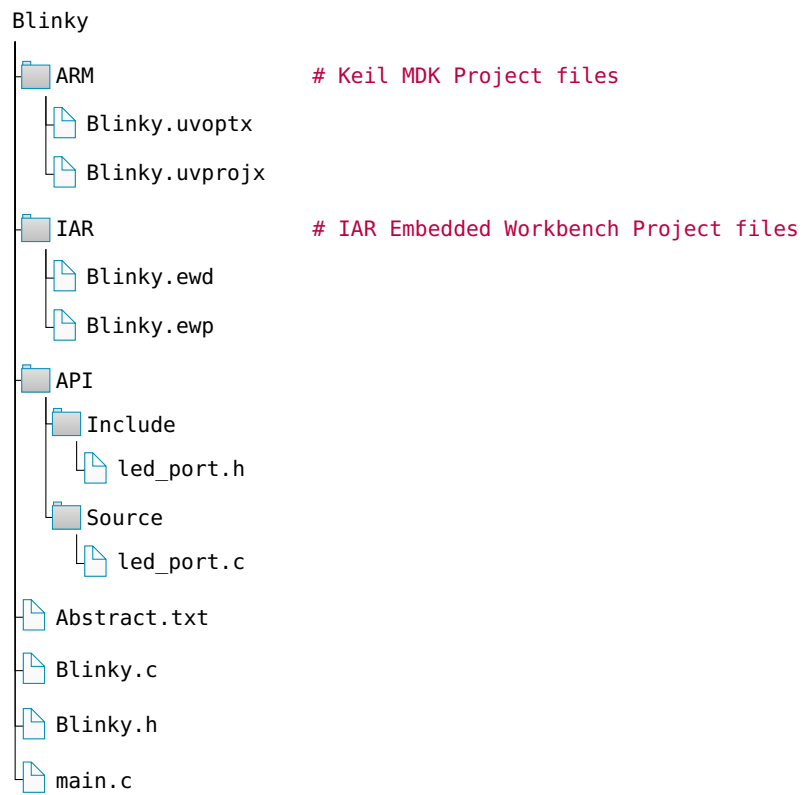


Figure 35: Blinky examples folder structure

### 7.2 Vio example tree

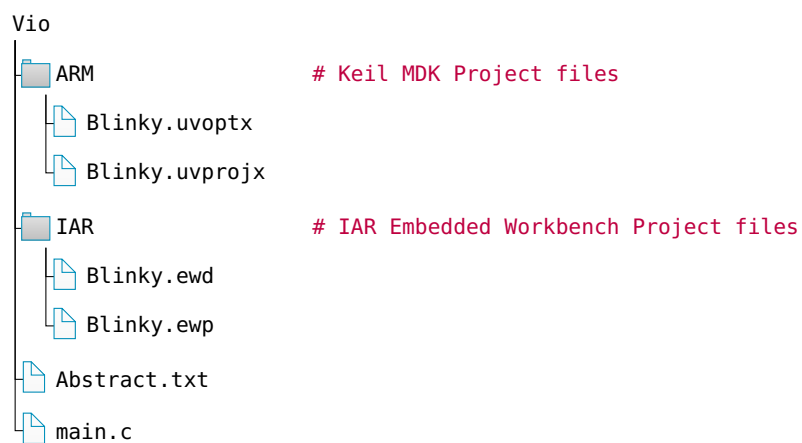


Figure 36: VIO examples folder structure