Table of Contents

1. Introduction

This document is intended to help users prepare a Windows system to connect to an Apollo Class MCU. The Ambiq Micro Apollo MCU is currently supported by 3 tool chains:

1. Eclipse(MARS)/gcc
2. Keil MDK
3. IAR EWARM

Before installing or using any of these tool chains, the user should first install the Ambiq Control Center, as discussed in Section 2 below. All of the examples shipped in the Ambiq Control Center come with source code as well as precompiled binaries from each of the tool chains. Even without a tool chain installed, one can run all of the example programs from the Ambiq Control Center by selecting the AMFLASH utility.

Once the Ambiq Control Center is installed, the user can then download the desired tool chain as discussed in the appropriate section below.

2. Installing and Using the Ambiq Control Center

The Ambiq Control Center installer provides most of the tools needed for connecting to an Apollo Class MCU. This includes USB drivers for the in-circuit debugger interface, and a custom version of OpenOCD, along with a few other utilities. This portion of the document will explain the process of installing the Ambiq Control Center.

2.1 Run the Installer

Along with your development hardware, you should have received an executable installer for Ambiq Control Center. You can obtain this installer by going to www.ambiqmicro.com/support. It will have the icon shown in Figure 1.

Figure 1: Installer Icon

After running the installer, you may receive a message with text similar to, “Do you want to allow the following program from an unknown publisher to make changes to this computer?” Answer Yes to this prompt.

The “AmbiqControlCenter” initialization dialog will first be shown. After a few seconds, the Welcome panel will appear, click Next to advance to the License agreement screen. See Figure 2.

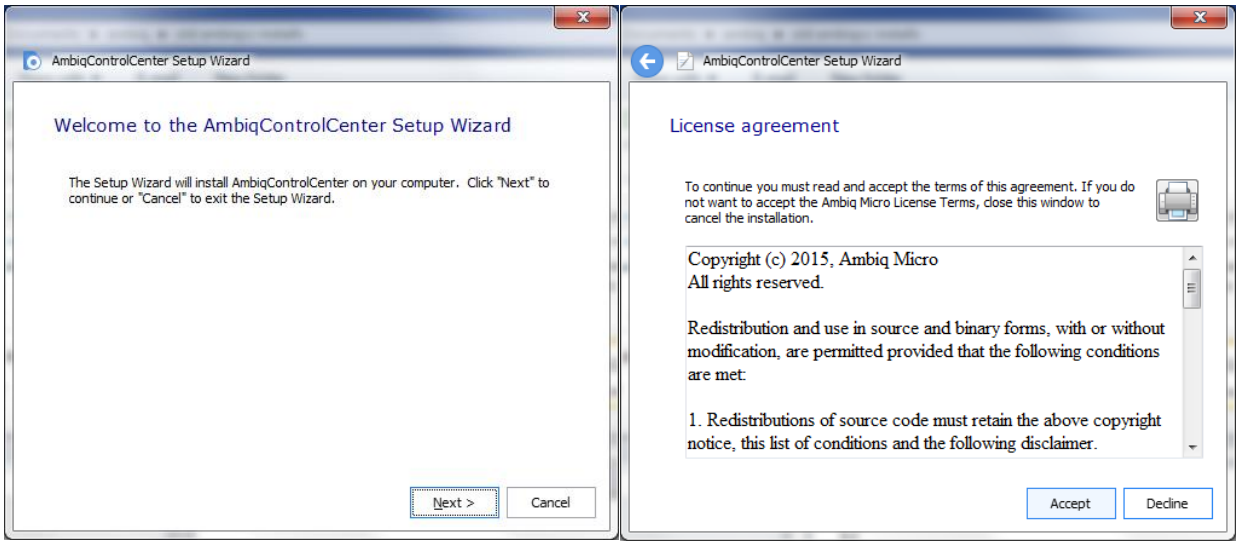


Figure 2: Welcome and License Agreement Screens

Before continuing, you will need to accept the terms of the license agreement.

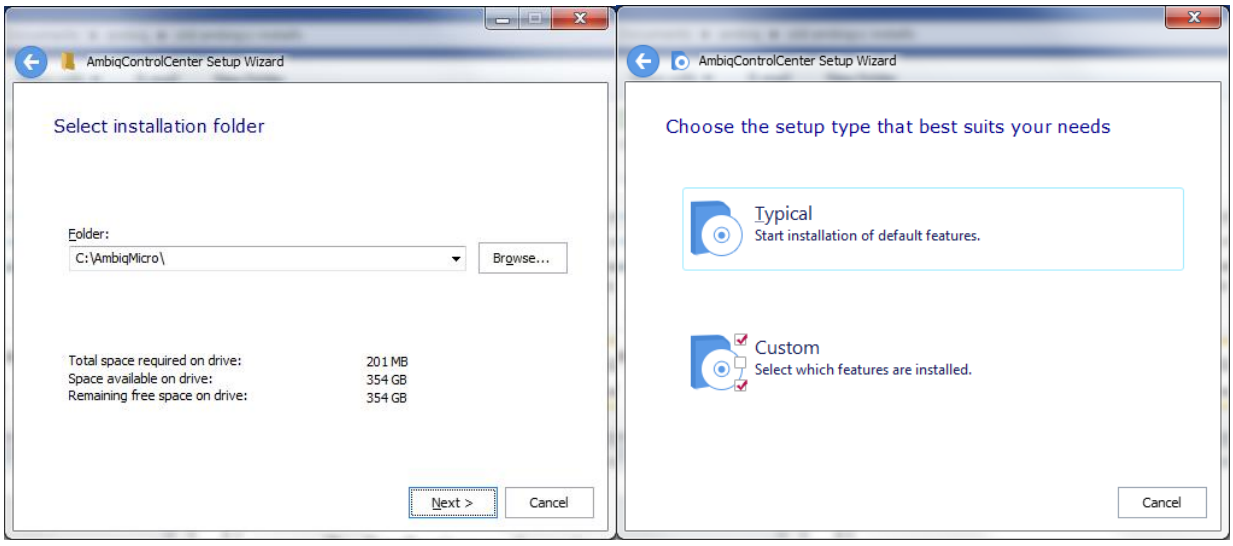


Figure 3: Installation Options

Keep the default “C:\AmbiqMicro\” as the installation folder, click Next to accept it. In the next panel, choose a “Typical” installation. This will install all of the necessary components for the AmbiqSuite to your PC. Note that the Eclipse and gcc based tool chain is installed separately and is covered in section 3,

Installing and Using the Eclipse/gcc Tool Chain on page 20.

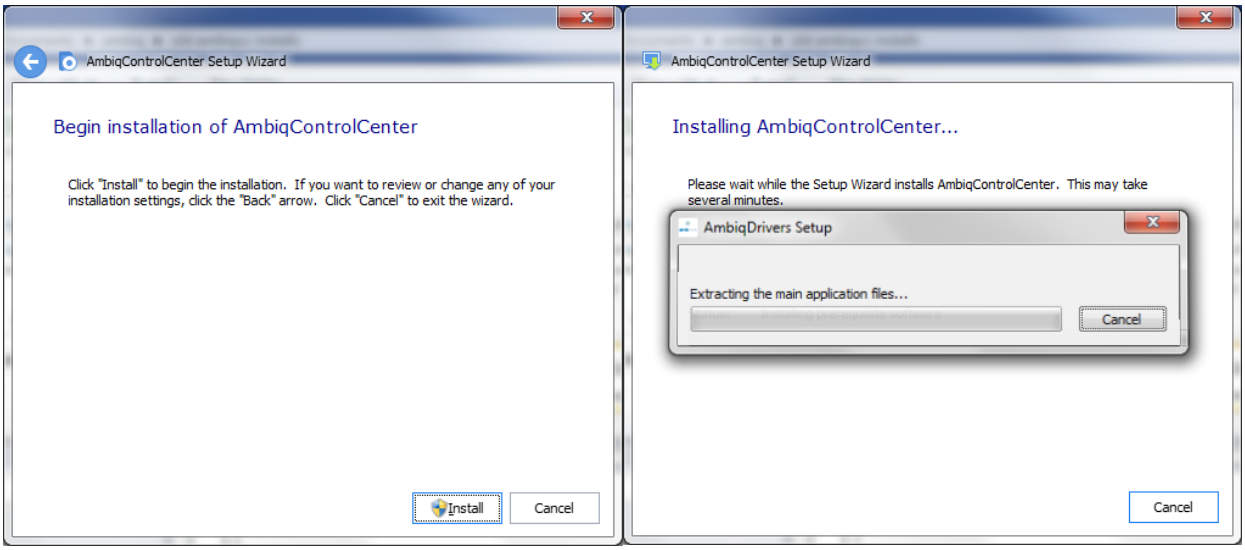
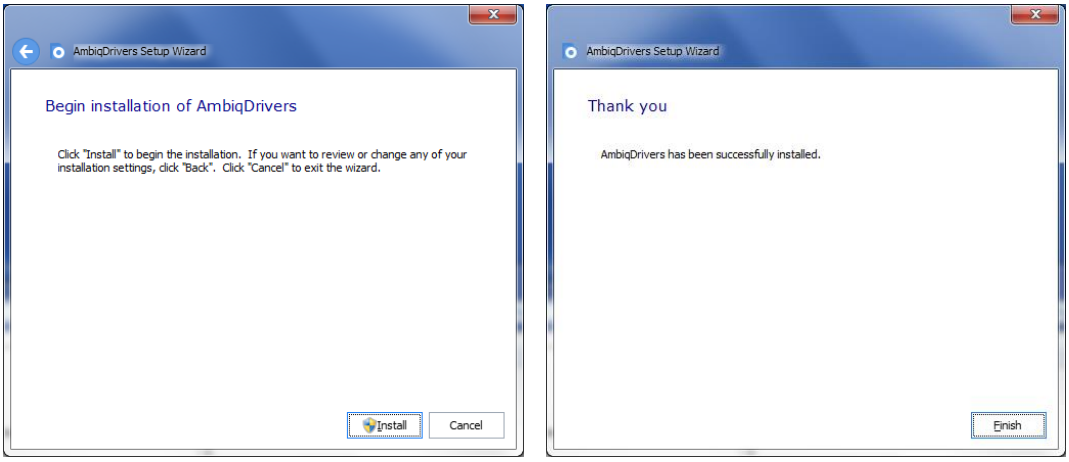


Figure 4: Installing Components

Next, click “Install”, and wait for the installation process to continue. A sub-installer for the drivers for the FTDI debugger interface on the EVK board will be started from the main installer at this point, displaying the following panels:



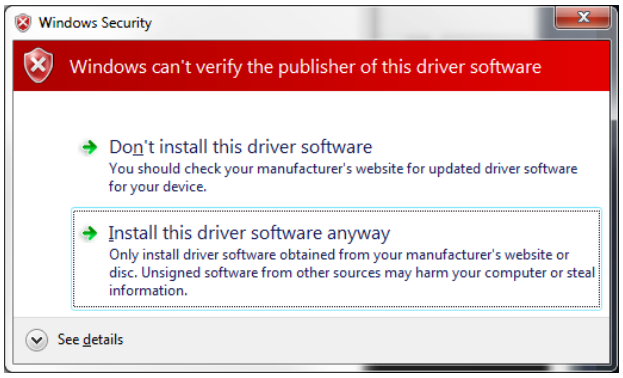


Figure 5: Installing FTDI drivers via AmbiqDrivers

This action installs two sets of drivers. The first is the standard FTDI drivers for the AmbiqMicro instance of the FT4232H or FT2232H device used as the debugger interface. The second set of drivers is the libusbK driver which is used by the openOCD daemon to control the serial wire debugger (SWD) interface pins on the Apollo MCU.

Please note that during the driver installation, particularly if this is the first time that the FTDI drivers have been installed on the target Windows system, a red dialogue may pop up warning that the driver being installed cannot be verified. In order to continue driver installation, you will need to click “Install this driver software anyway”.

Once the driver installation is finished, the main Ambiq Control Center installer will continue to its completion.

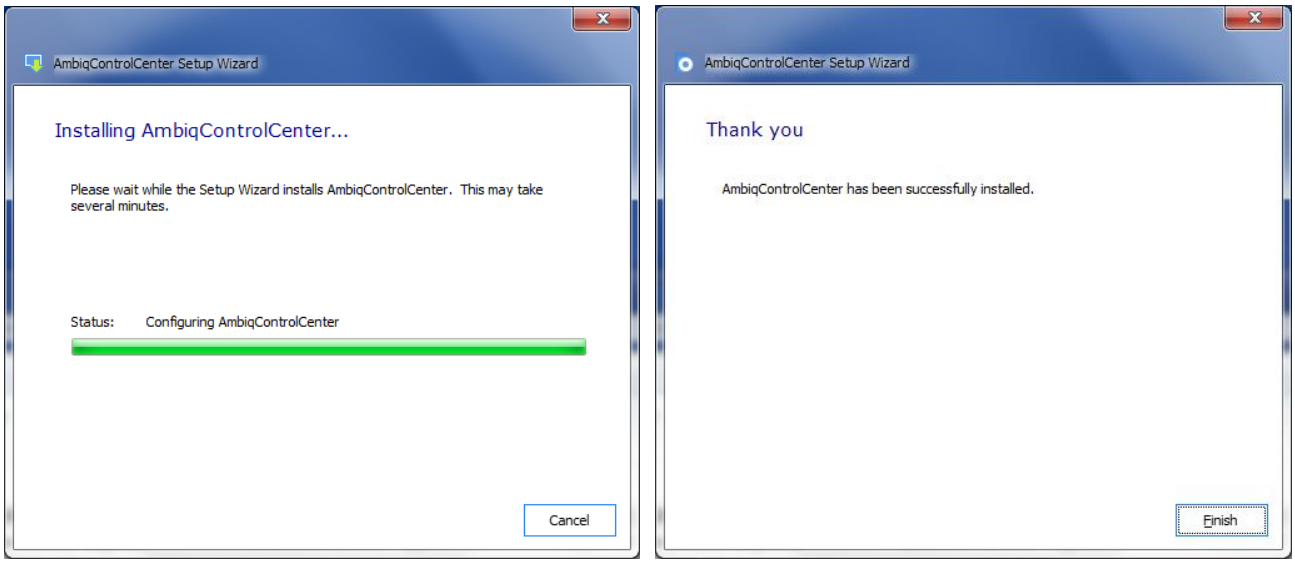


Figure 6 Finishing the Ambiq Control Center Installation

Click on the Finish button. At this point, you will have Ambiq Control Center and FTDI drivers fully installed on your PC at the selected location (C:\AmbiqMicro by default).

2.2 A Word about the Apollo EVK Board Stack and It's Integrated Debugger Interface

While the tool chains and examples depicted in this document can be used with any board with a suitable debugger connection and peripheral chips, we focus our discussions herein upon the Apollo EVK board as shown in Figure 7. The EVK consists of a two board stack. The top board shown in the figure is the EVK base board which has the Apollo MCU mounted dead center on the board. The lower board, partially visible in the figure, is a sensor board that gives this EVK stack the "personality" of a sensor platform suitable for modeling a wearable device or a sensor hub.

As seen in the figure, the Apollo EVK base board has two methods of connecting to the debugger in one the supported tool chain IDEs to the EVK:

1. Standard 10-pin ARM ULINK2 style connector.
2. USB connection facilitated by an FTDI FT4232H chip on the Apollo EVK base board.

Figure 7 shows the USB connection in use and the jumpers set appropriately to support the FTDI based connection. One should refer to the Apollo EVK documentation for more details on jumper settings.

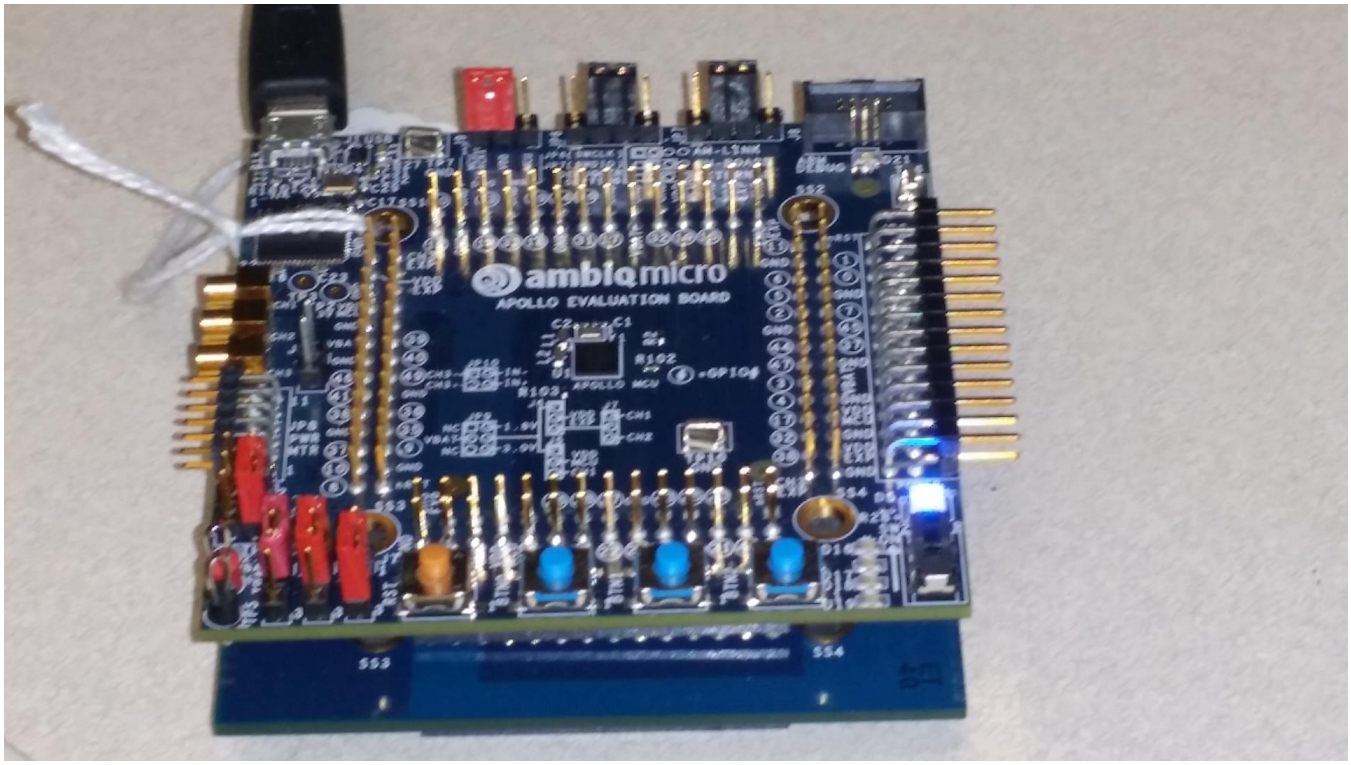


Figure 7 Apollo EVK Board

The Apollo EVK base board is supported by the openOCD debug daemon which controls the SWD pins via FT4232H channel A. The openOCD daemon presents a gdb connection to a debugger in the Eclipse IDE and the IAR IDE. Thus Eclipse and IAR eWARM use this mechanism to connect their IDE/debugger applications to the Apollo MCU. Ambiq Micro provides an AGDI driver for the Keil MDK IDE connection to the Apollo MCU's SWD pins. The Apollo MCU SWO pin is connected to FT4232H channel C and presents as a Windows Com Port. These details are discussed further in the sections below.

The AMFLASH Utility, discussed in section 2.4, uses the openOCD daemon to control the Apollo EVK base board. It allows one to download and run precompiled binaries of the example programs right out of the box.

2.3 Using the CoreSight 10 pin Debugger Socket

One can use the ARM 10-pin CMSIS connector to debug embedded firmware on the Apollo MCU by using a Keil ULINK2 USB probe with the Keil MDK IDE or an IAR iJET USB pod with the IAR EWARM IDE. Change the EVK jumpers to the configuration shown in the figure:

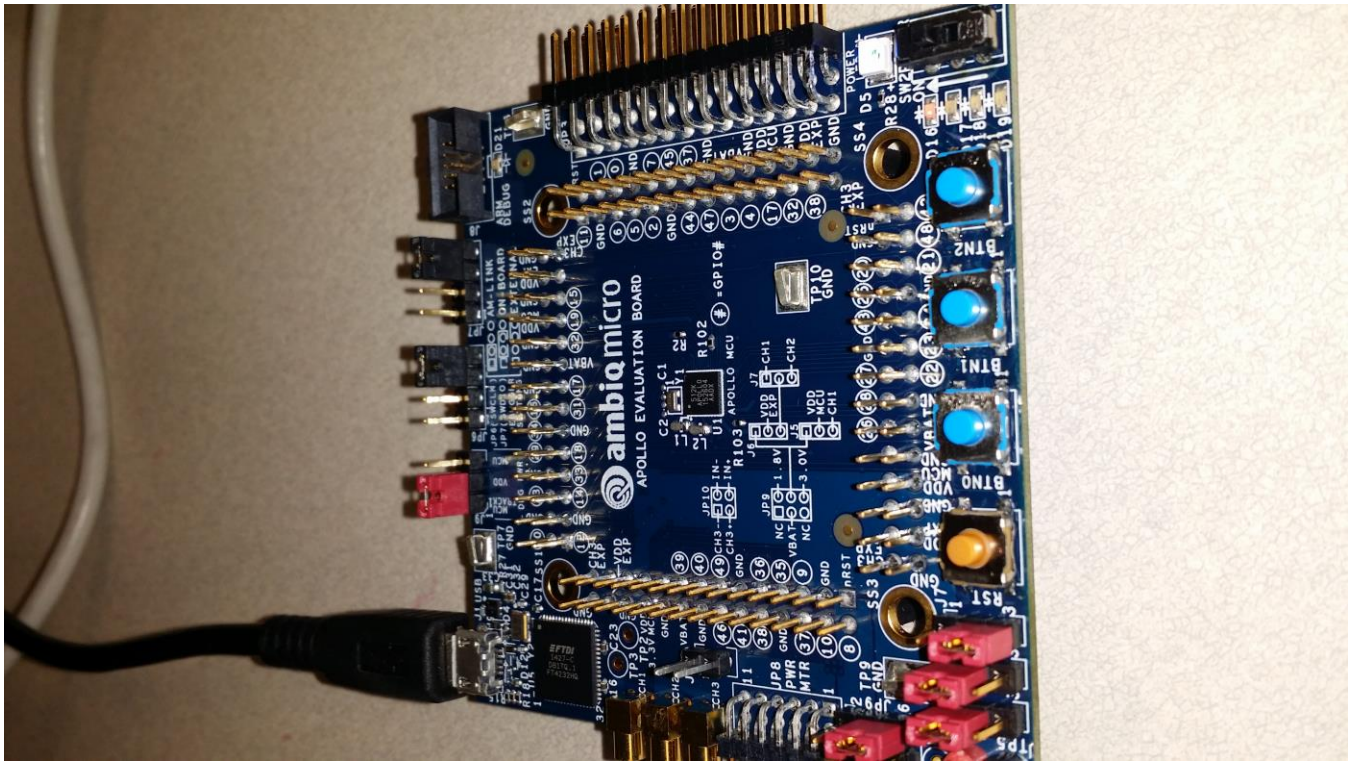


Figure 8 Setting EVK Jumpers for Use with ULINK2 or iJET

2.4 Using the Ambiq Control Center

Everything that one needs to experience the Apollo MCU and to run the provided example programs can be accomplished with the Ambiq Control Center and the AMFLASH utility without downloading or installing any of the 3 tool chains. Once you have the Ambiq Control Center installed, go to the utilities menu and select the AMFLASH utility.

First Check that the drivers installed correctly by launching openOCD from within the Ambiq control Center.

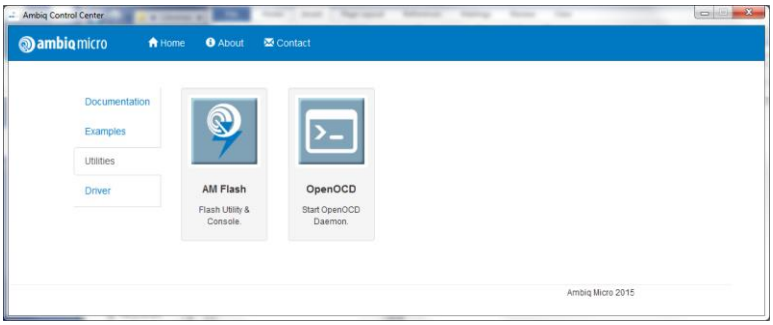


Figure 9 Launch the openOCD utility from with Ambiq Control Center

You should see a windows command shell pop up like the following figure.

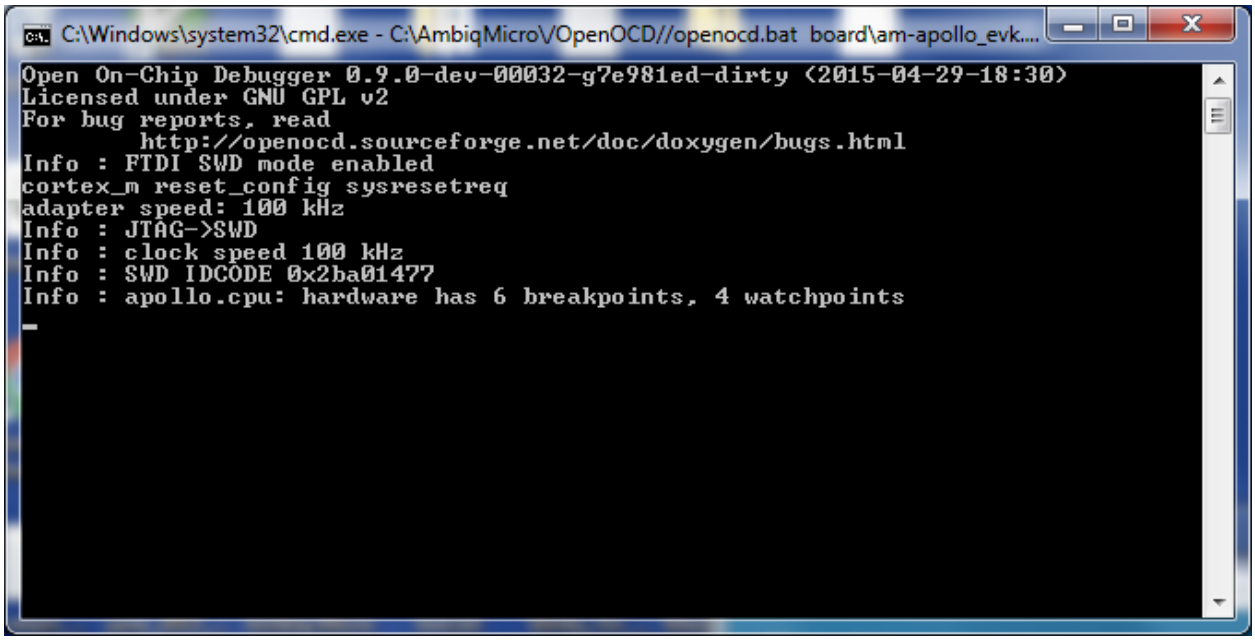


Figure 10 OpenOCD Console View

Make sure that it says 6 breakpoints and 4 watch points. If you see these numbers then the drivers are functioning correctly for debugger access to the Apollo MCU over the FTDI chip. If the OpenOCD command shell looks correct, skip the next paragraph to continue with using the AMFlash utility.

If this does not succeed, first try a power cycle of the Apollo EVK and restart OpenOCD. If that doesn't resolve it, try retargeting the drivers using the driver control panel in the Ambiq Control Center, as shown in Figure 11. Try using the "Install libusbk" radio button. This will reinstall the drivers needed by openOCD and AMFlash.

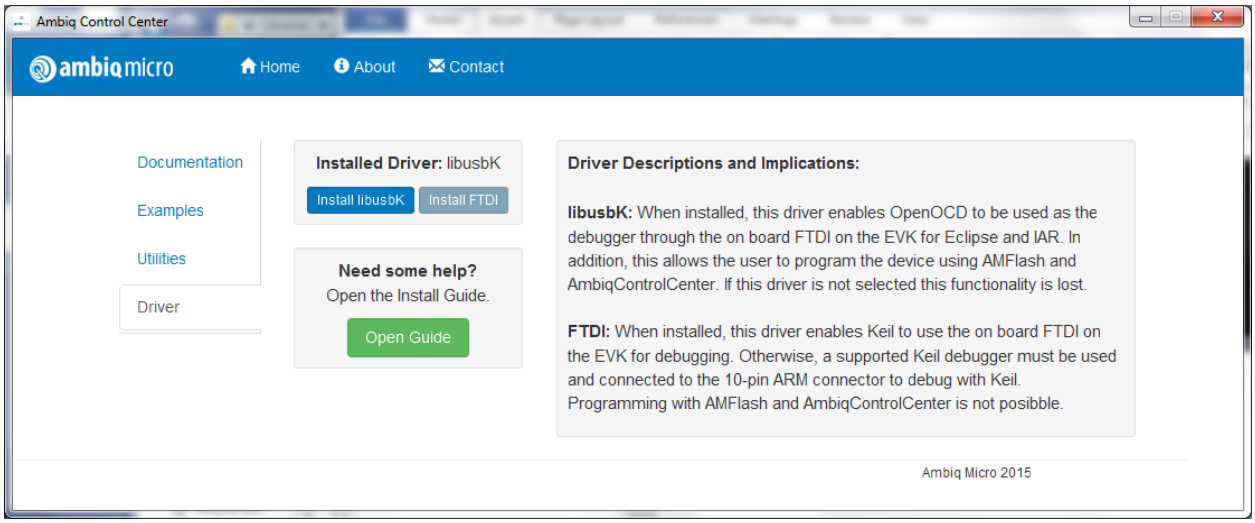


Figure 11 Driver Selection Panel

Once OpenOCD is working as expected, AMFlash will be able to operate properly. Since AMFlash quietly starts the openOCD daemon as needed, the openOCD command shell will need to be closed, which will kill openOCD. AMFlash may not function properly if openOCD is already running. However, you will need to start openOCD in a command shell if working with Eclipse (see also section Start OpenOCD).

2.4.1 Ambiq Control Center Documentation

The Ambiq Control Center gives you access to lots of installed documentation using the Documentation button, as follows:

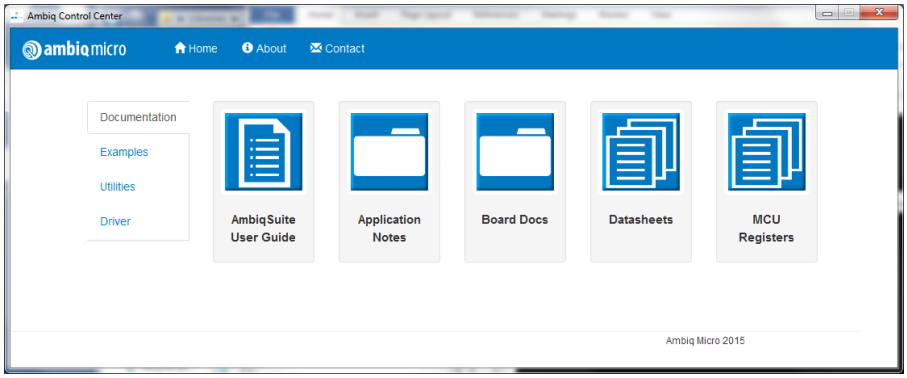


Figure 12 Ambiq Control Center Documentation Options

The AmbiqSuite User Guide contains the doxygen extracted documentation for the APIs.

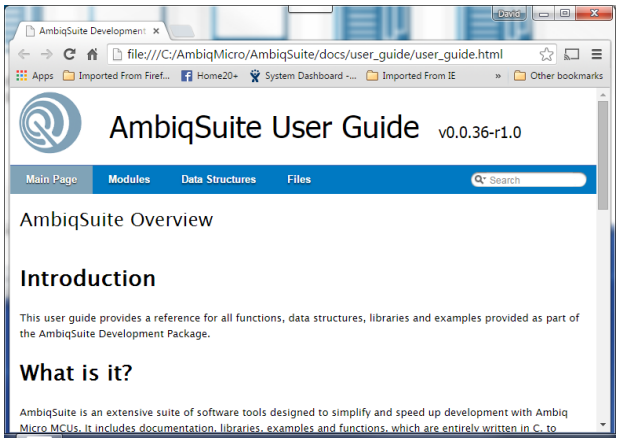


Figure 13 AmbiqSuite User Guide

The Ambiq Control Center also provides quick access to several application notes, including this Windows debugging application note (Quickstart Guide). It contains a similar app note explaining how to setup and then use a Linux platform for debugging the Apollo MCU. It also contains an app note explaining how to activate the AGDI driver for Keil that enables Keil debugging using the FTDI chip on the Apollo EVK Base board with needing a separate ULINK2 USB debug probe.

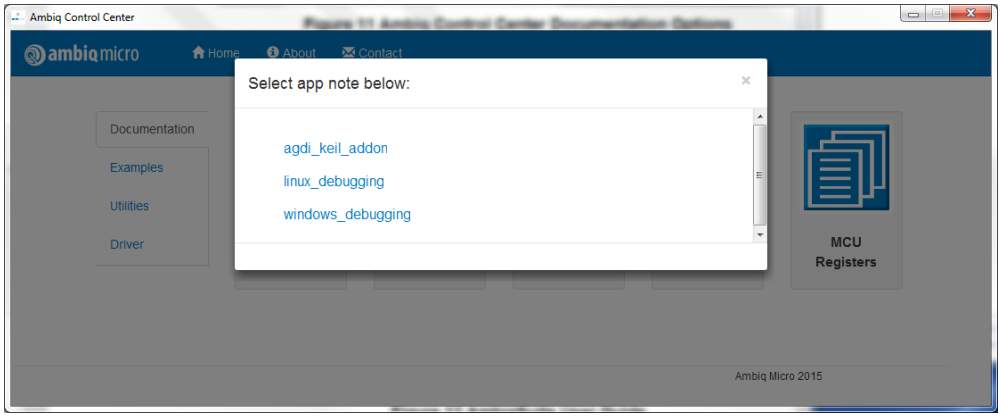


Figure 14 Ambiq Control Center Application Notes

The board documentation accessible from within the Ambiq Control Center includes the user's guide for the Apollo EVK Base board as well as schematics for all of the boards in the Apollo EVK stack. The current Data Sheet for the Apollo MCU can also be easily accessed from within the Ambiq Control Center.

Finally, detailed Apollo programmable register documentation can be easily accessed from within the control center by clicking on the MCU documentation button. Your browser will be automatically loaded with an index page that lists every peripheral block in the MCU as well Cortex system registers, see Figure 15.

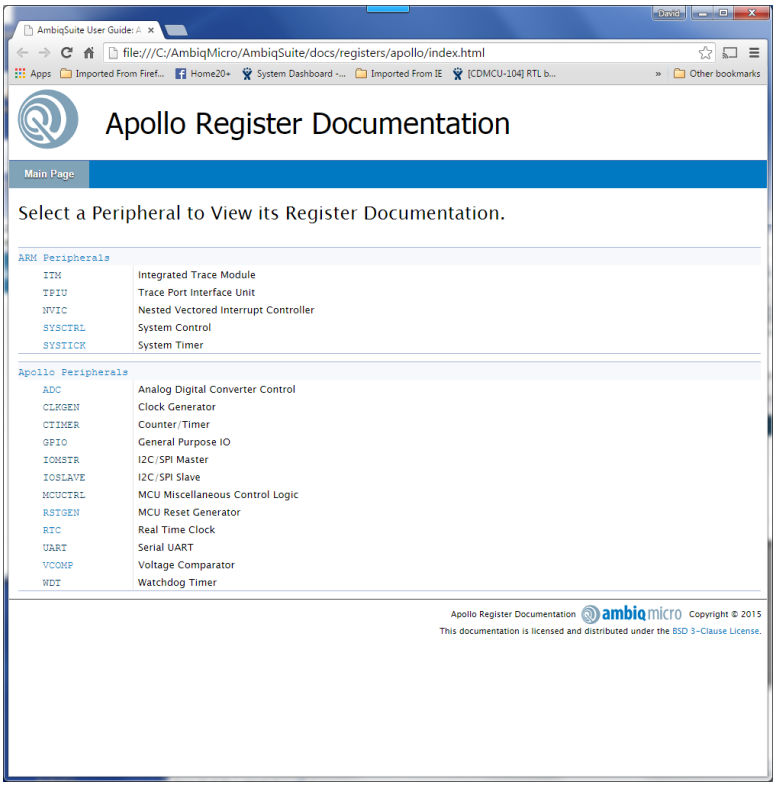
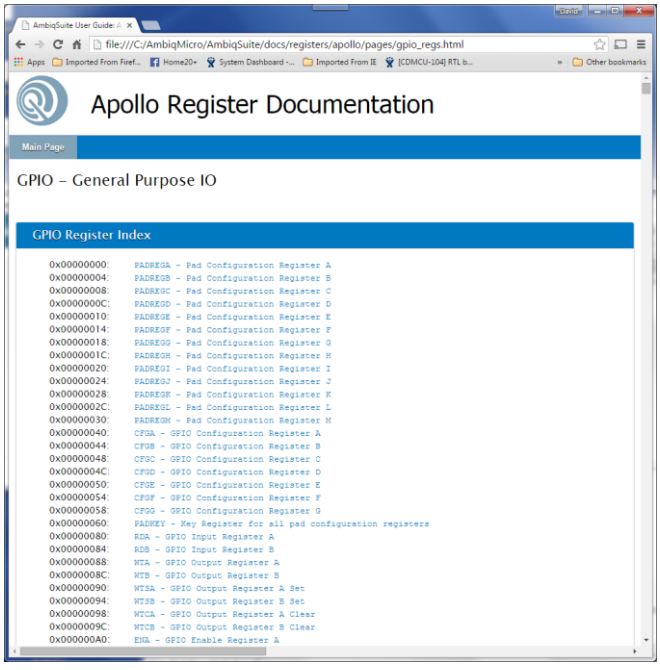


Figure 15 Register Documentation Block Selection

Selection of the GPIO block for example will give you page listing all of the registers in the GPIO block. Clicking on one of the register names will take you to a page detailing the definition of the bit fields in that register. Selecting the GPIO CFG A register for example will provide a page like the one shown in Figure 16.



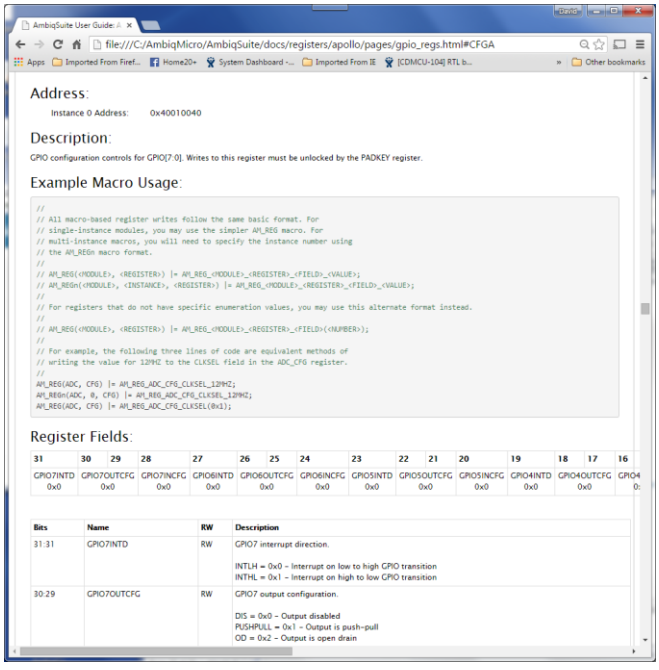


Figure 16 GPIO Register Selection

2.5 A Word about the Structure of the AmbiqSuite Contents

The AmbiqSuite ships with 8 directories at its top level as shown in Figure 17.

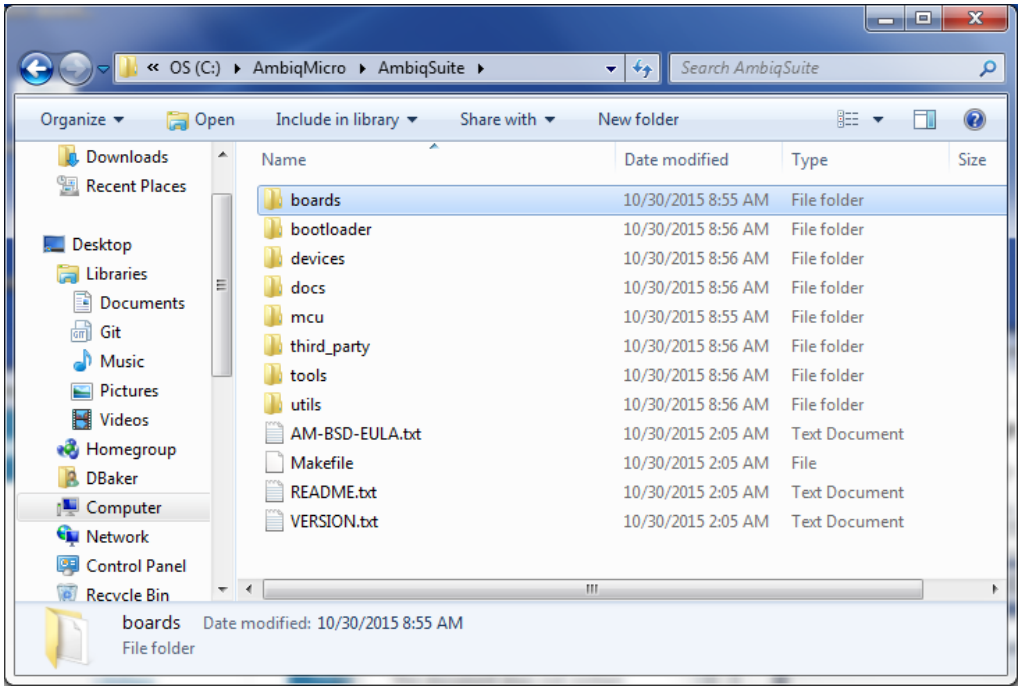


Figure 17 Top Level Directory of AmbiqSuite

Selecting the boards directory shows us that we can navigate to directories supporting any of the 3 board configurations: EVK base board stand alone, EVK base board plus EVK sensor board or a 3 board combination consisting of the EVK base, EVK sensor and the EVK Bluetooth Low Energy radio board (BTLE). For this discussion, we will dive into the EVK base board directory, see Figure 18.

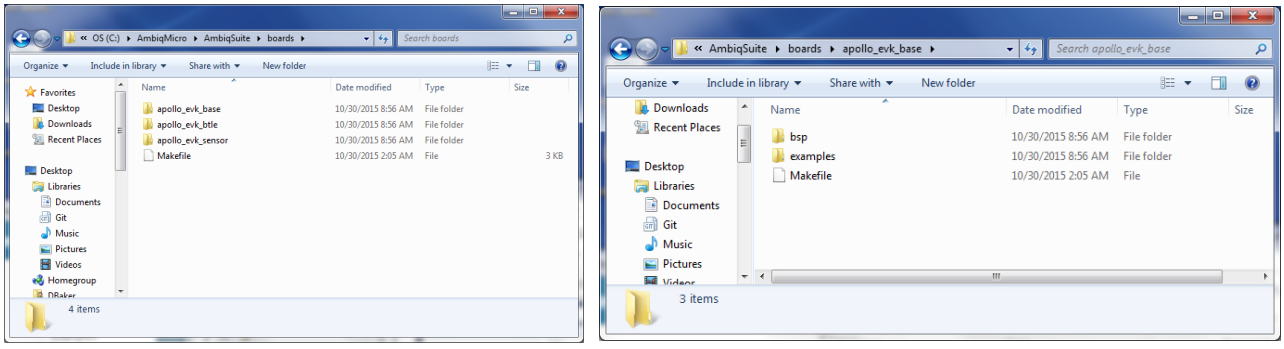


Figure 18 Diving into a Board Directory, Such as the one for the Apollo EVK Base

The Board Support Package (BSP) directory contains code that is specific to the target board configuration. It contains files that assign uses to pins on the Apollo MCU and assign devices to the appropriate I/O master. The contents of the BSP directory are different for every board configuration supported. Finally, if we dive into the

examples directory for the EVK base board we see a long list of examples provided with the AmbiqSuite. None of the examples in this directory requires sensors or Bluetooth.

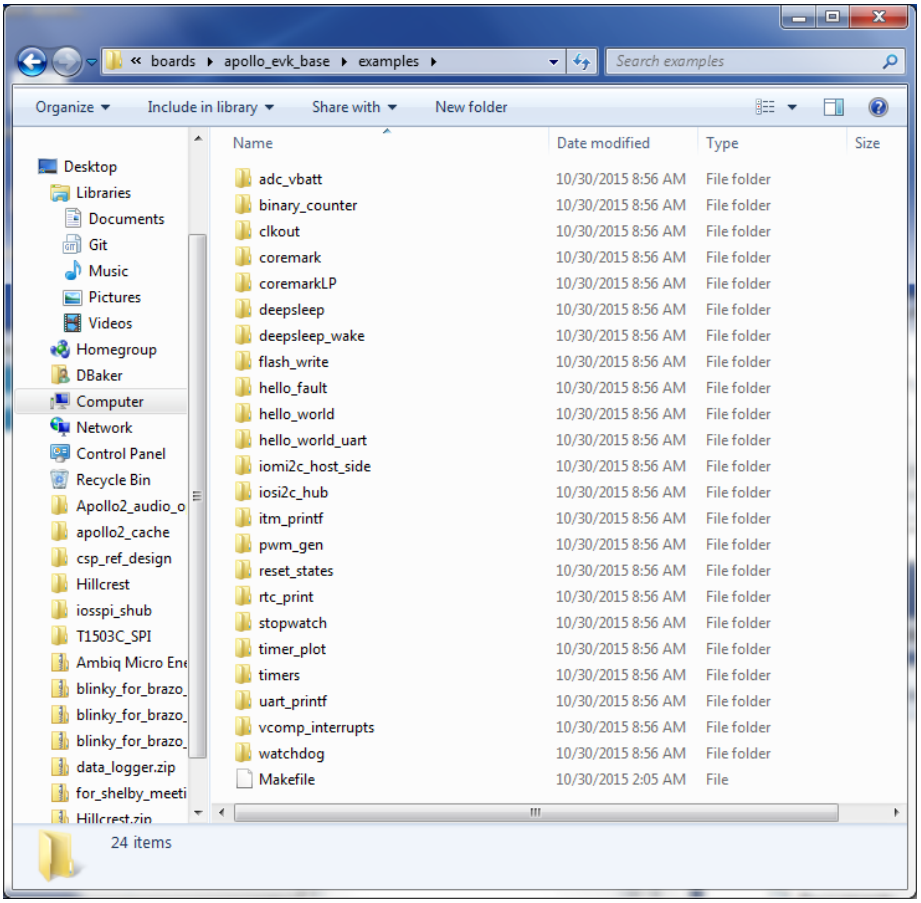


Figure 19 Contents of the EVK Base Board Examples Directory

The AmbiqSuite ships with precompiled binaries for all of its examples. All of these binaries can run from directly within the Ambiq Control Center by selecting the Example button from the main panel see Figure 20.

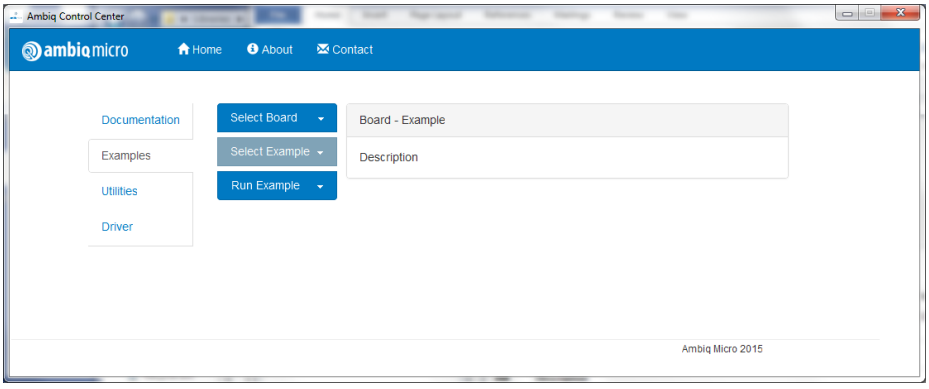


Figure 20 Example Selection Dialog

If you select the apollo_evk_base board and the hello_world example then you see the panel of Figure 21

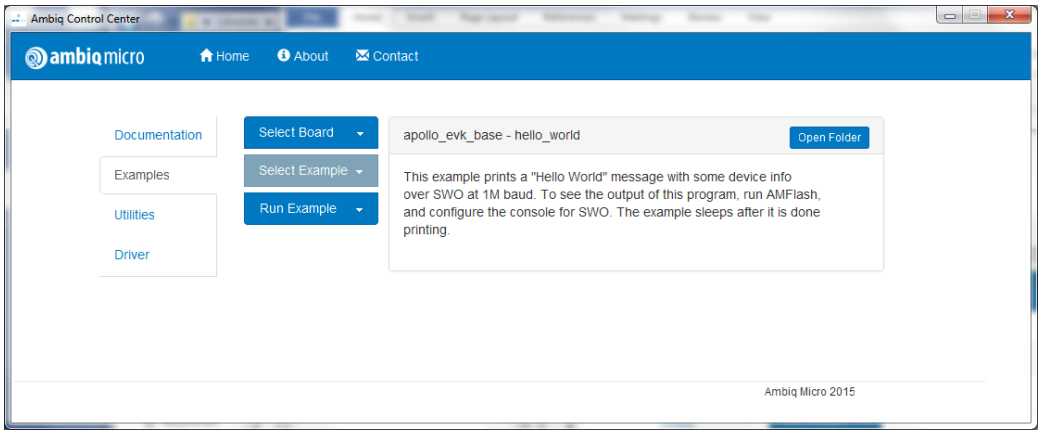


Figure 21 Example Selection

The example description text is derived from the doxygen markup in the example source file. A button will take you directly to the directory containing the selected example, i.e. hello_world in this case. Pressing the run button causes the precompiled binary for the example to be downloaded onto the EVK and then run using the openocd debugger interface. Note that the run button is a pull down allowing you to select the specific binary compiled with the desired one of the three tool chains (gcc, Keil, IAR). This selection option is very convenient when you are experimenting with the ULP Benchmark example.

2.6 Using the AMFLASH Utility from Within the Ambiq Control Center

Everything that one needs to experience the Apollo MCU and to run the provided example programs can be accomplished with the Ambiq Control Center and the AMFLASH utility without downloading or installing any of the 3 tool chains. Once you have the Ambiq Control Center installed, go to the utilities menu and select the AMFLASH utility as shown in Figure 22.

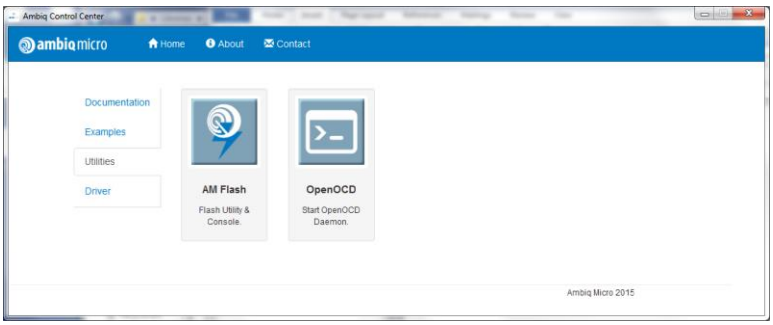


Figure 22 Launch the AM Flash utility from with Ambiq Control Center

The AMFlash utility is a standalone program that can also be launched from an icon that was added to your desktop during the installation process. Either way that you launch it, you see the startup screen of

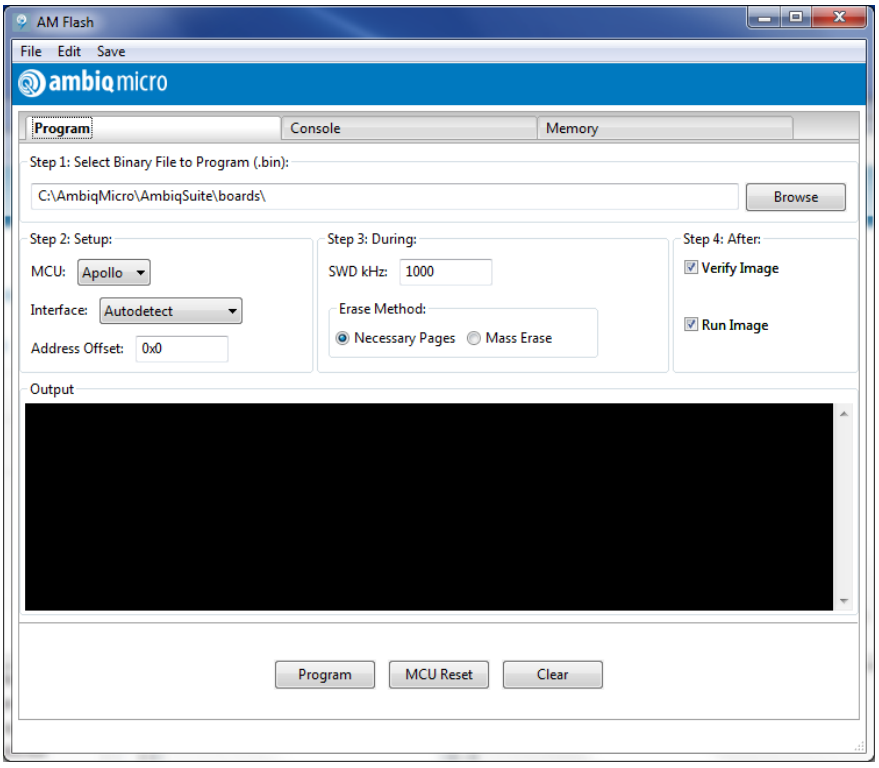


Figure 23 AMFlash Utility

There are 3 main panels in this utility: one for downloading a programming to the Apollo MCU; one for displaying console output from things like debug printf statements and one for erasing all of the integrated FLASH on the Apollo MCU. In order to run the hello_world example we will need to be able to see console output so we will go set that up first. Push the button on the Consoletab.

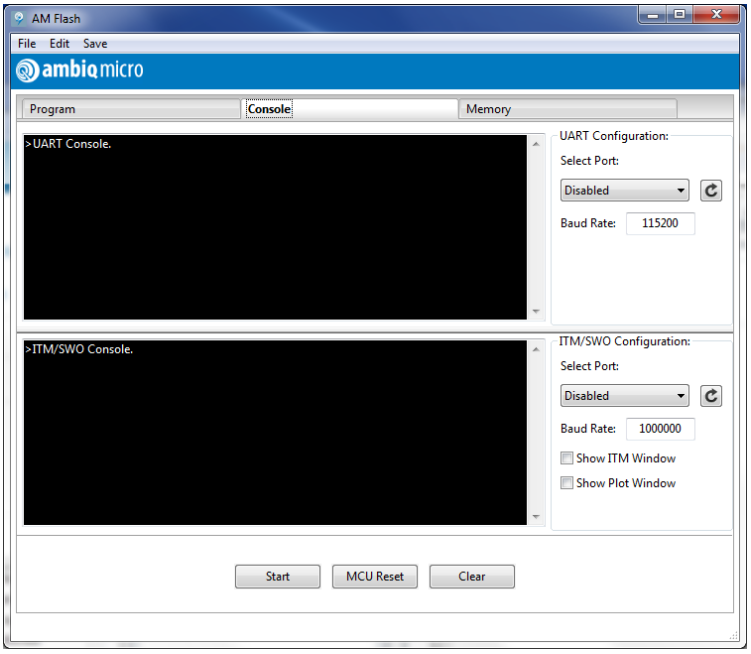


Figure 24 AMFlash Utility Console Panel

The console panel gives access to a large number of viewing options, starting with displaying the output from Apollo UART, if it is configured to use GPIO[35] and GPIO[36]. These pins are hard wired to FTDI FT4232 channel D on the Apollo EVK base board. Usage of this UART is infrequent because the UART is usually dedicated to talking to the Bluetooth radio device. We will discuss this console much later. First we will show how to activate the ITM/SWO (serial wire output) display using the ITM/SWO console.

2.6.1 Connecting the Com Port for the SWO output pin to AMFLASH

To use this console we need to configure it to use the appropriate COM port to collect the SWO packets from the Apollo MCU as shown here:

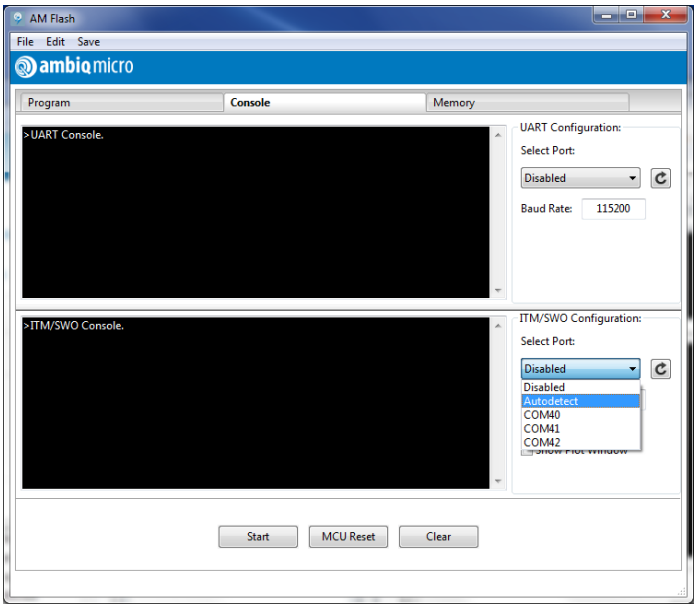


Figure 25 Autodetecting the SWO COM Port

Selecting the autodetect option and clicking Start causes AMFlash to search the available COM ports for the one connected to the SWO pin. When it finds it, it makes it the selected port and starts the ITM/SWO console. The Ambiq print utilities use ITM stimulus register 0 to implement a printf library. We will use this console to see the hello_world example's output.

2.6.2 Downloading and Running the Hello World Example

Downloading and running a program like the hello world example in the EVK base examples directory is very easy with AMFlash. Go to the Program panel of AMFlash and browse to the hello world example directory. Browse into the eclipse_gcc\bin directory and click on hello_world.bin, as shown in Figure 26.

At this point you can click the Program button, causing the hello world example to be downloaded into the Apollo flash in preparation for running it. You will see a flash progress bar as it down loads. When it finishes, all you need to do to run it is to issue a reset to the Apollo MCU.

Click over to the console panel and you will see the Hello World output. Easily rerun the example by clicking the "MCU Reset" button.

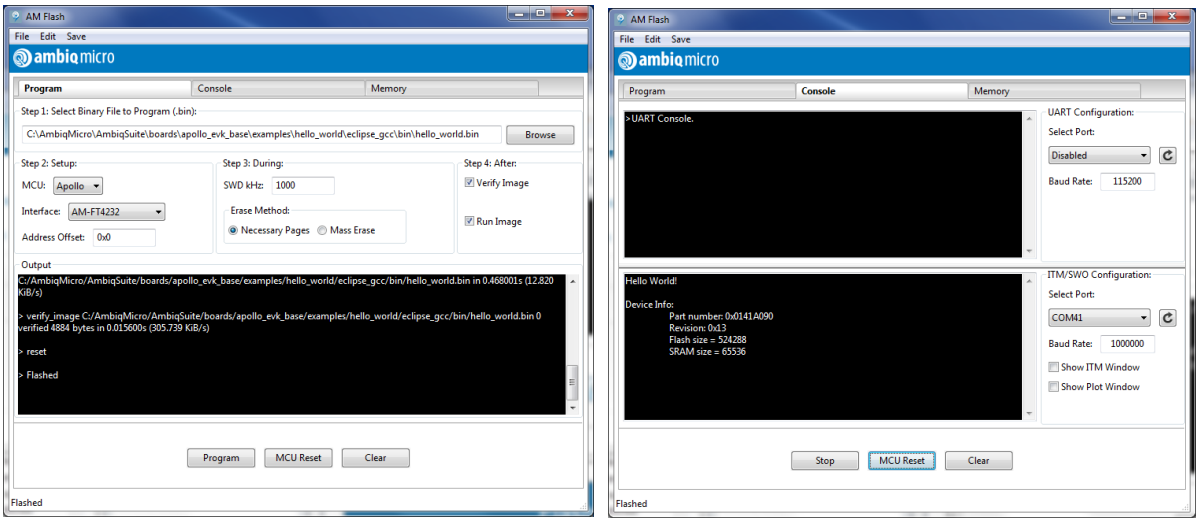


Figure 26 Downloading and Running the Hello World Example

2.6.3 Show ITM Plot Window and the Show ITM Window Buttons

AMFlash can extract data samples from values written to ITM registers 24 through 27 and a number of the examples in the am_evk_sensor board package use this facility to plot the output of gyros, accelerometers and magnetometers on that board. Once the ITM port is selected and started for the console tab, one simply clicks on the Show Plot Window to get a plot such as this one in Figure 27. In addition, one can elect to see the most recently written value to each of the ITM stimulus registers as well. If you are running an example that emits plot samples you will the values in stimulus registers 24 through 27 changes as the plot updates.

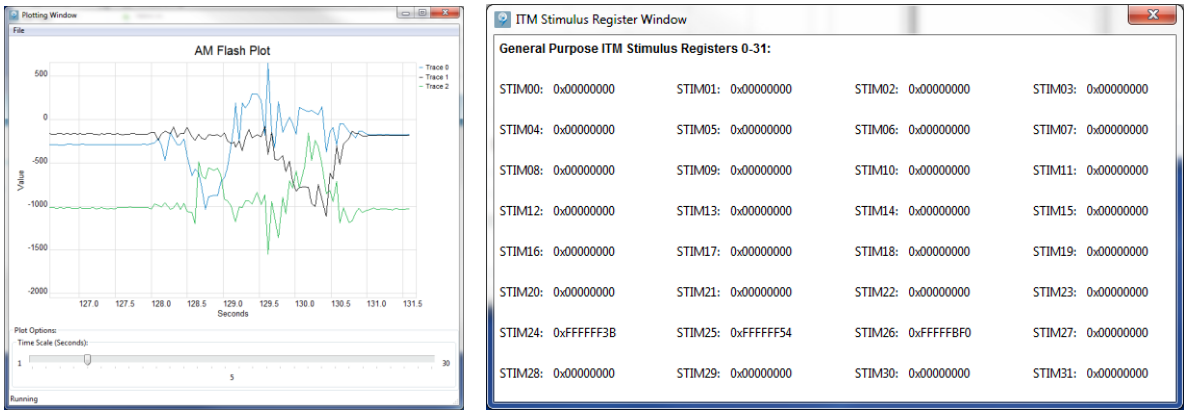


Figure 27 AMFlash Plot Window and ITM Stimulus Register Window

2.6.4 Performing a Mass Erase or Recovery on an Apollo EVK

Sometimes it becomes necessary to completely erase a program to recover convenient debug control of a part. This may be necessary when programs like the watchdog timer example are continuously timing out and resetting the MCU.

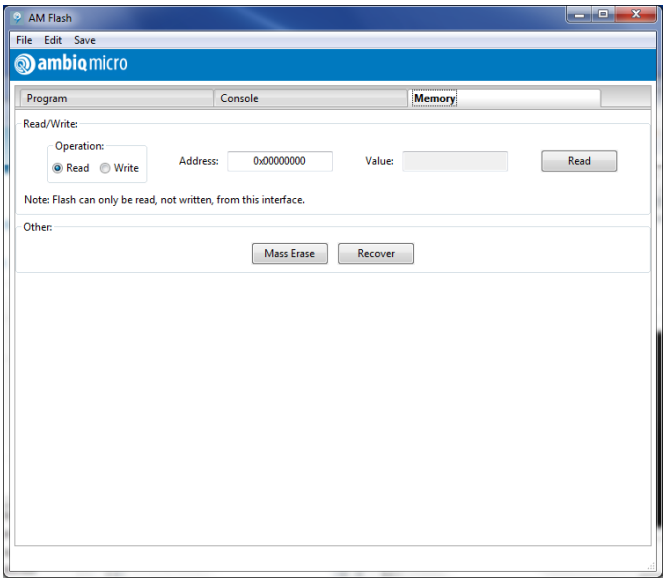


Figure 28 Mass Erasing the Apollo MCU Flash from AMFlash Utility

3. Installing and Using the Eclipse/gcc Tool Chain

The Ambiq Debug Tools installer provides the GNU gcc compiler and Eclipse (Mars version) IDE that can be used for Apollo software development and debug. This portion of the document will explain the process of installing the Ambiq Debug Tools environment.

3.1 Installing the Eclipse/gcc Tool Chain

3.1.1 Prerequisites

Before installing Ambiq Control Center, make sure that your PC has a copy of the most recent version of the Java Runtime Environment installed. This is required for the Eclipse IDE, and is available as a free download from the Oracle website.

3.1.2 Install Eclipse/gcc

After launching the Ambiq Micro DebugTools installer, the following dialogue may be encountered; for example if the correct Java run time environment is not already installed on the target system. Click Next to continue with the Prerequisites Wizard, then install whatever recommendations the installer makes.

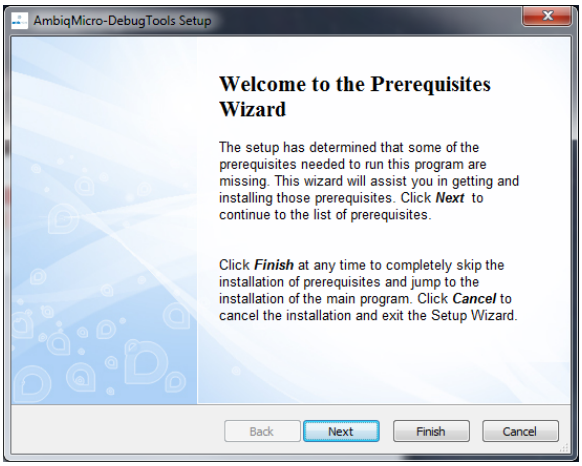


Figure 29 Prerequisites Wizard

Many of the initial panels of the Debug Tools installer are similar to those for Ambiq Control Center including a Welcome (select Next), License agreement (read and select Accept), and even a Typical or Customer panel. On the latter panel, select Typical to install the Eclipse IDE and GNU tools. Then select the default installation folder and click Next.

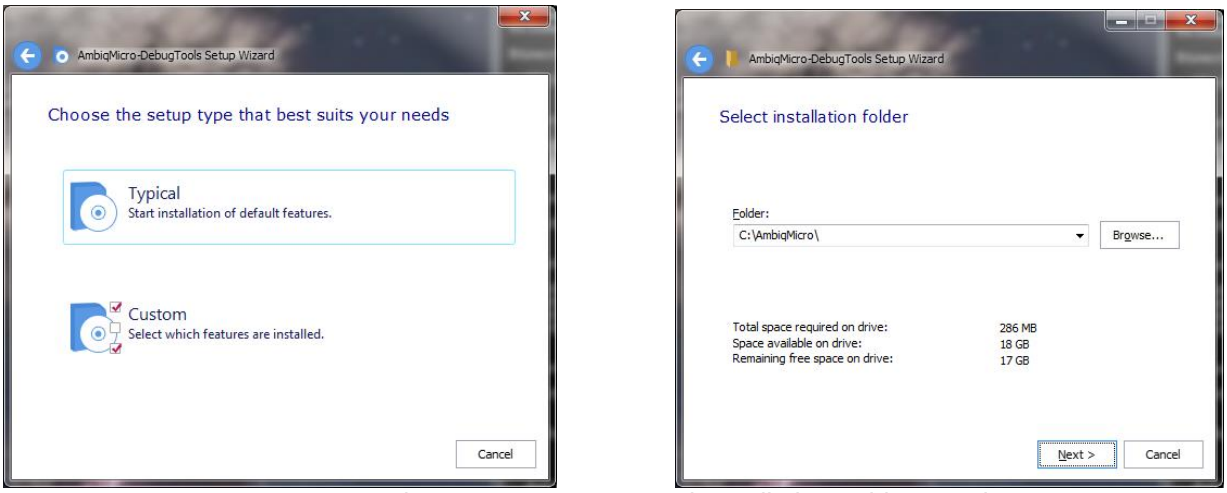


Figure 30 Setup Type and Installation Folder Panels

Finally, the Begin Installation panel will appear, click Install.

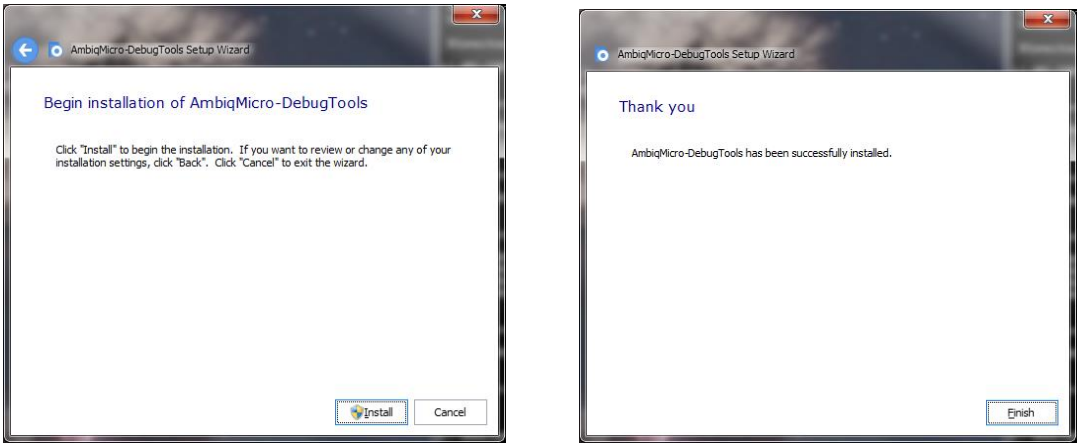


Figure 31 Begin Installation and Finish Panels

Click Finish to exit the installer once the installation is complete.

3.1.3 Starting Eclipse and Initial Screens

Once installed, an Eclipse shortcut will exist on the desktop. Using the shortcut, start Eclipse Mars and resize the window as desired. The very first time Eclipse is started, a welcome screen will be displayed which contains various helpful links including Eclipse overview, Tutorials, Samples, What's New, and Go to the workbench. Access the desired links and tutorials, then click "Go to the workbench", at which point you'll be at the following window.

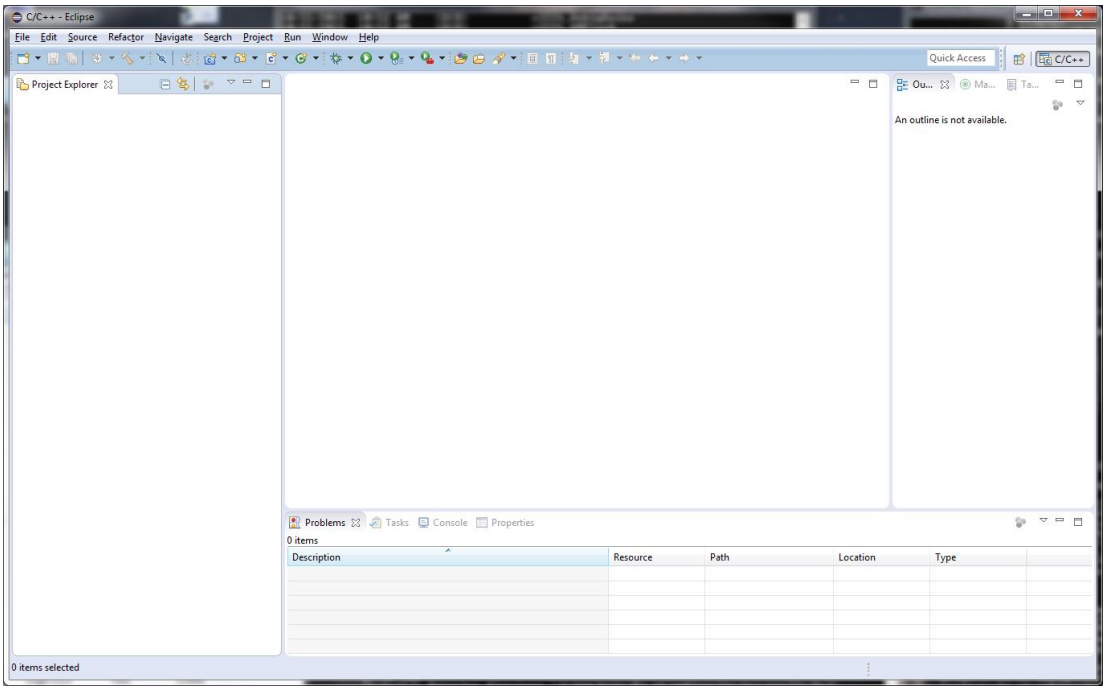


Figure 32 Initial Eclipse Window After Installation

3.2 Importing an Eclipse Project and Debug

3.2.1 Import and Build a Project

Importing an existing AmbiqSuite project is straightforward. First click ‘File’ (or alternatively, right click in the Project Explorer area) and select ‘Import’, as shown in Figure 34. In the window that pops up, expand ‘General’, and select ‘Existing Projects into Workspace’ as shown in Figure 35, then click Next. On the final screen, referencing Figure 33Figure 34, browse to or copy the path to the project, click “Deselect All”, deselect “Copy projects into workspace”, then select the desired project(s) (in this case, hello_world) and select ‘Finish’, at which point the selected project(s) is loaded into the Project Explorer window of Eclipse.

To build the imported project: select the desired project, click the hammer icon in the toolbar (or alternatively press ‘Ctrl+B’ on your keyboard). The console will output some status messages. The project is ready for debugging once it has been successfully built.

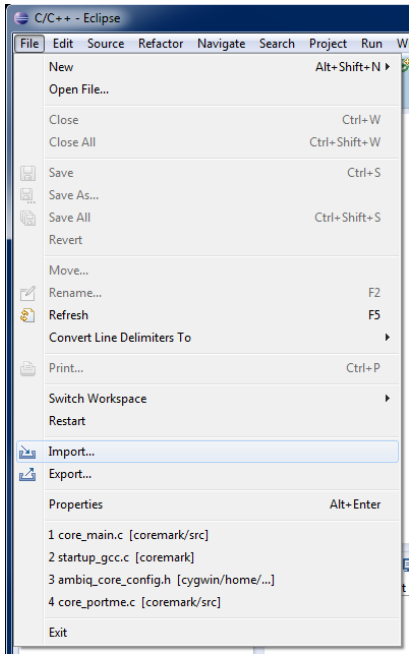


Figure 34: Select 'Import'

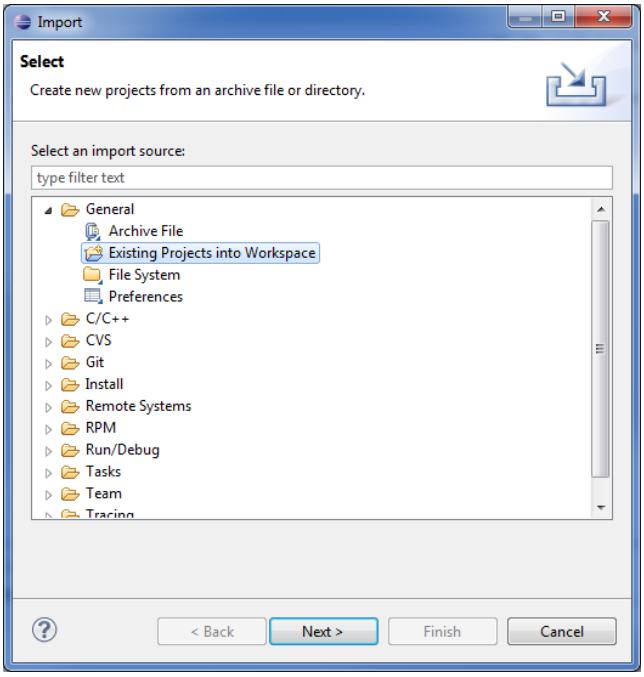


Figure 35: Select 'Existing Projects into Workspace'.

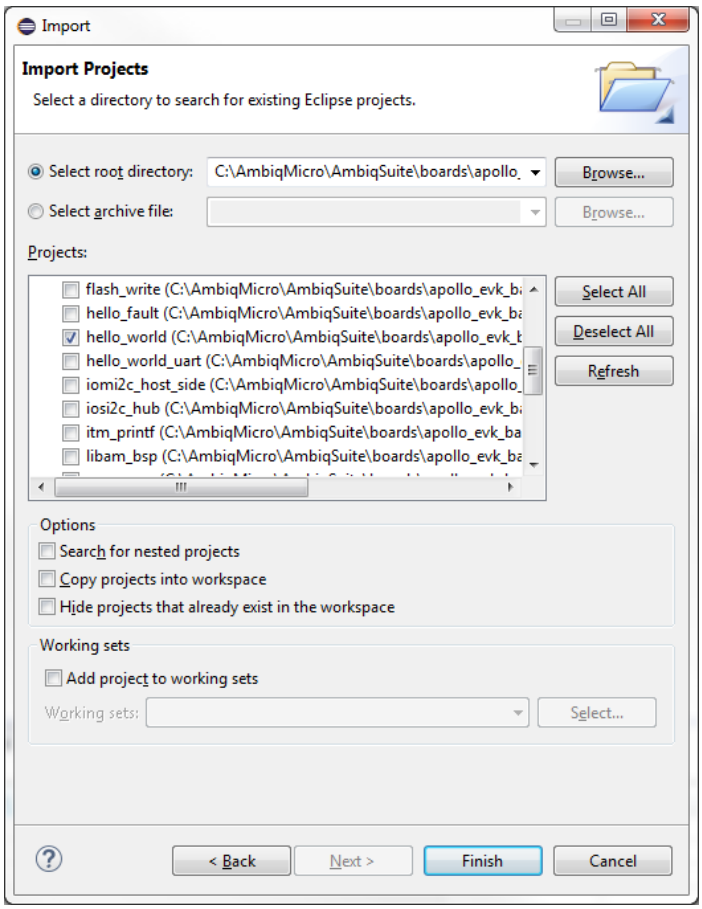


Figure 36: Browse to the example project

3.2.2 Start OpenOCD

In order to debug within Eclipse, OpenOCD needs to be running. Therefore (as also previously discussed in Using the Ambiq Control Center) OpenOCD must be manually started to connect Eclipse to the running process using GDB. The easiest way to start OpenOCD is to use Ambiq Control Center.

Figure 37: Ambiq Control Center

First, start Ambiq Control Center.

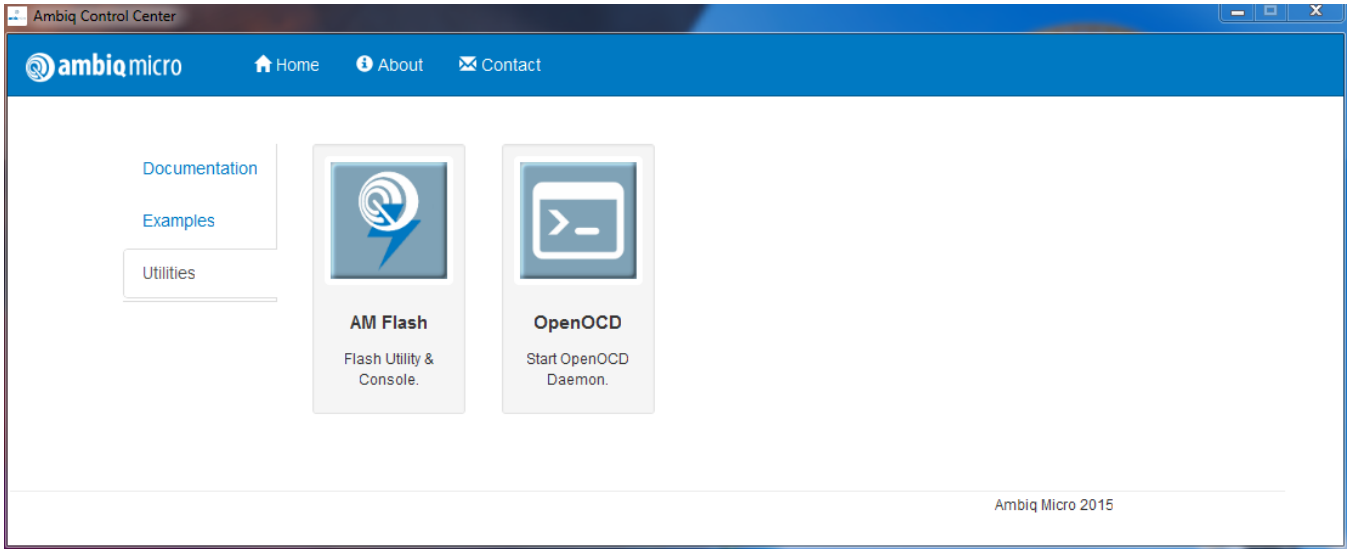
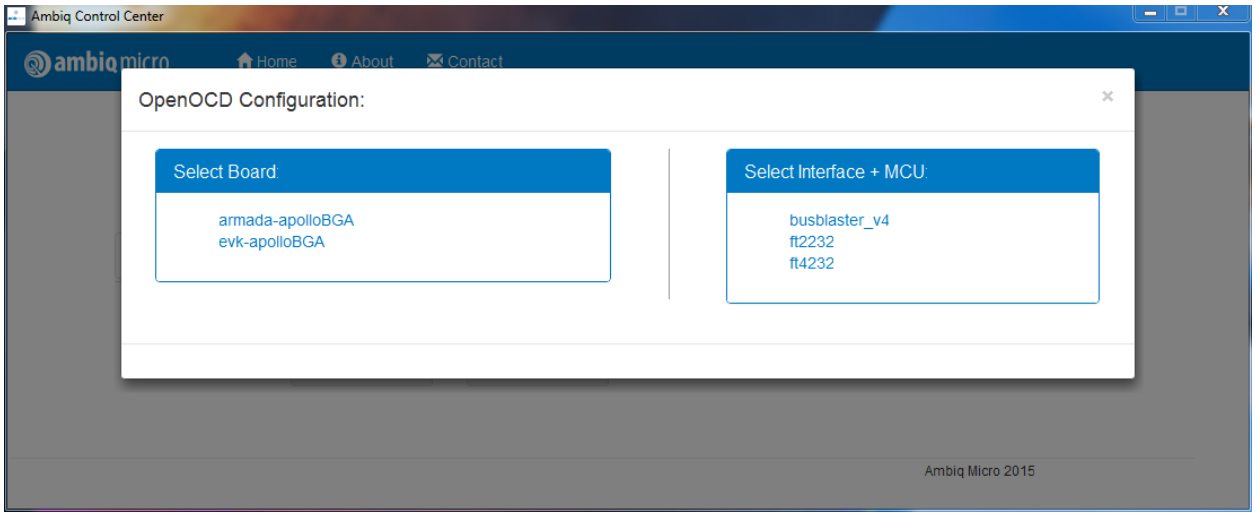


Figure 38: Ambiq Control Center

Then, open the “Utilities” tab, and click “OpenOCD” to start the OpenOCD Daemon.



Select your board from the list on the left. This should open a new window with information about the debug link to the Apollo device. Check to make sure your output looks like Figure 39. If your OpenOCD output is different, make sure that your Apollo device is plugged in and powered on, then close the window, and try again.

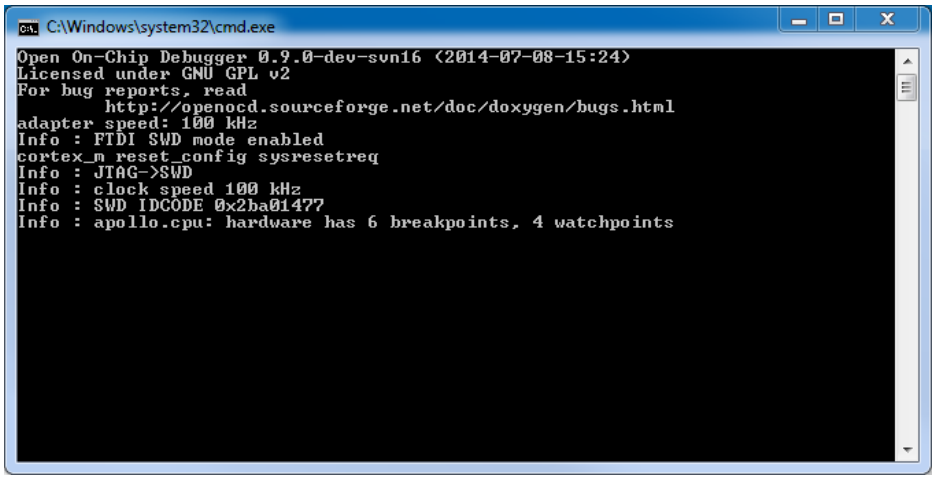


Figure 39: OpenOCD Started

3.2.3 Debug

Now that OpenOCD is running and waiting for a GDB connection, we can connect to it. To do that, click the down arrow next to the debug button as shown in Figure 40. Then click on “hello_world” (this name will change based on the name of your project). This will start GDB Hardware Debugging and take you to the Debug window as shown in Figure 44.

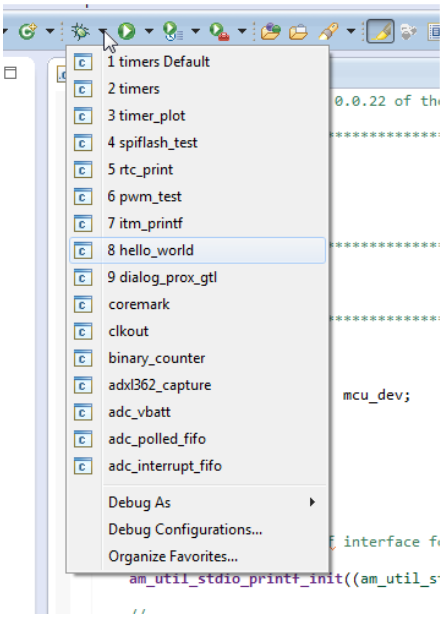


Figure 41: Debug Menu

Occasionally the drop down menu will fail to populate, especially on the first “import” operation in a new eclipse workspace. If this happens, you can also start the debugger by clicking “Debug Configurations” as shown in Figure 42, selecting your project in the “Debug Configurations” dialog, and clicking the “Debug” button as shown in Figure 43.

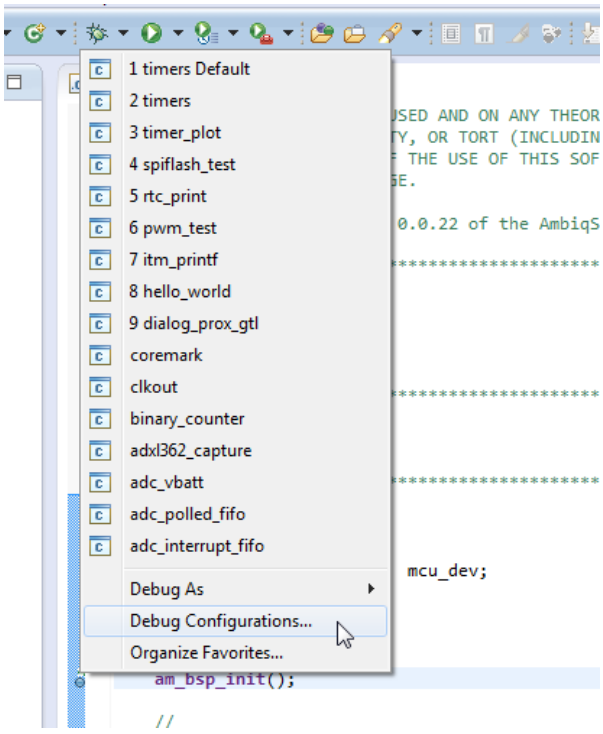


Figure 42: Click "Debug Configurations"

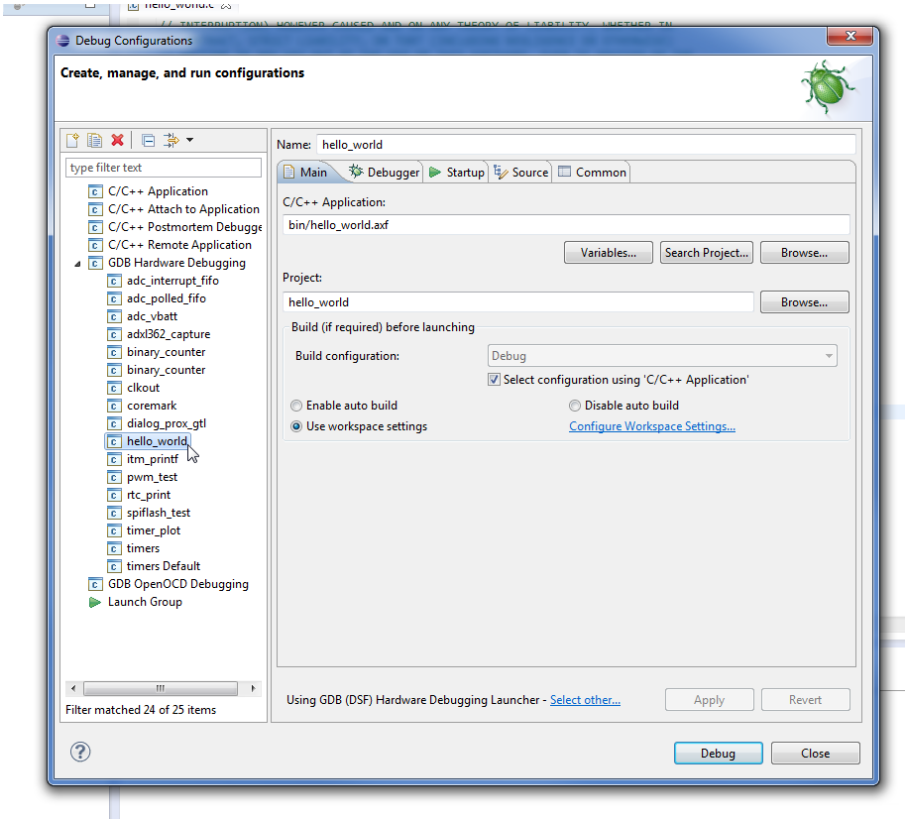


Figure 43: Debug Configurations Dialog

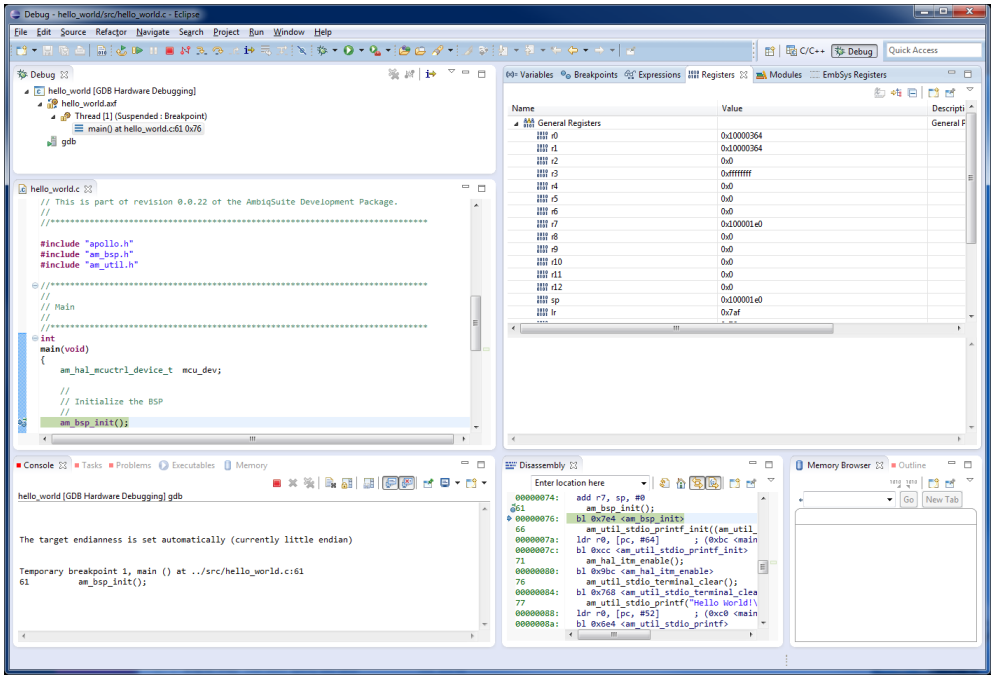


Figure 44: Eclipse Debug Window

You are now debugging! Set breakpoints, view memory & registers, run/halt or perform any debugging functions you are used to.

4. Build and Debug using Keil

Note: You should have Keil (minimum version v5.14) with the Ambiq Micro pack installed on the target PC.

If you haven't already done so, go to the Keil MDK download page at https://www.keil.com/demo/eval/arm.htm. Download the MDK and install it as directed by the Keil documentation. Use the Keil MDK pack installer and install the pack for the Ambiq Micro MCUs. For the Apollo EVK, you will be using the APOLLO512BGA device. If you will be using the Keil ULINK2 debug probe then install it and its drivers now. Connect the ULINK2 to the Apollo EVK Base board and change the jumpers as described in the Apollo EVK board user's guide. If you would like to use the Ambiq Micro supplied AGDI driver for the EVK instead of a ULINK2, then refer to section 2.3, "Using the CoreSight 10 pin Debugger Socket" on page 8.

4.1 Debugger Setup

Ambiq EVK boards ship with an on-board debugger that can be used with Keil without a separately purchased debugger probe (e.g. ULINK2). Follow instructions in "AM AGDI Keil Supplemental User guide" to setup the debugger and then return to section 4.3

4.2 Open a Project and Build

- Open an existing Keil project "binary_counter" from Ambiqsuite.

  The default location for Ambiqsuite project "binary_counter" is":

  "C:\AmbiqMicro\AmbiqSuite\boards\apollo_evk_base\examples\binary_counter\keil"

- Double click on the uVision Project file as shown in Figure 42 to open this projects using Keil

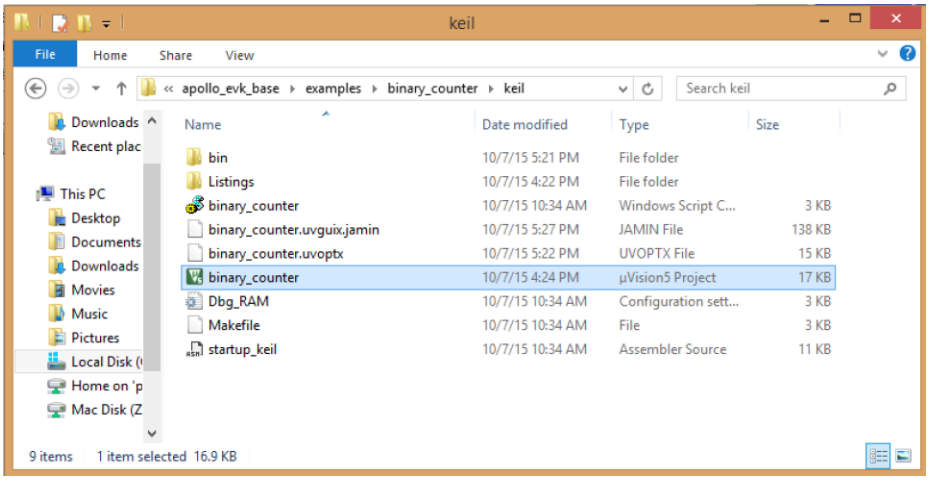


Figure 42: Keil Project folder for example "binary_counter"

- Build the projects at Project→Build Target as shown in the Figure 43.

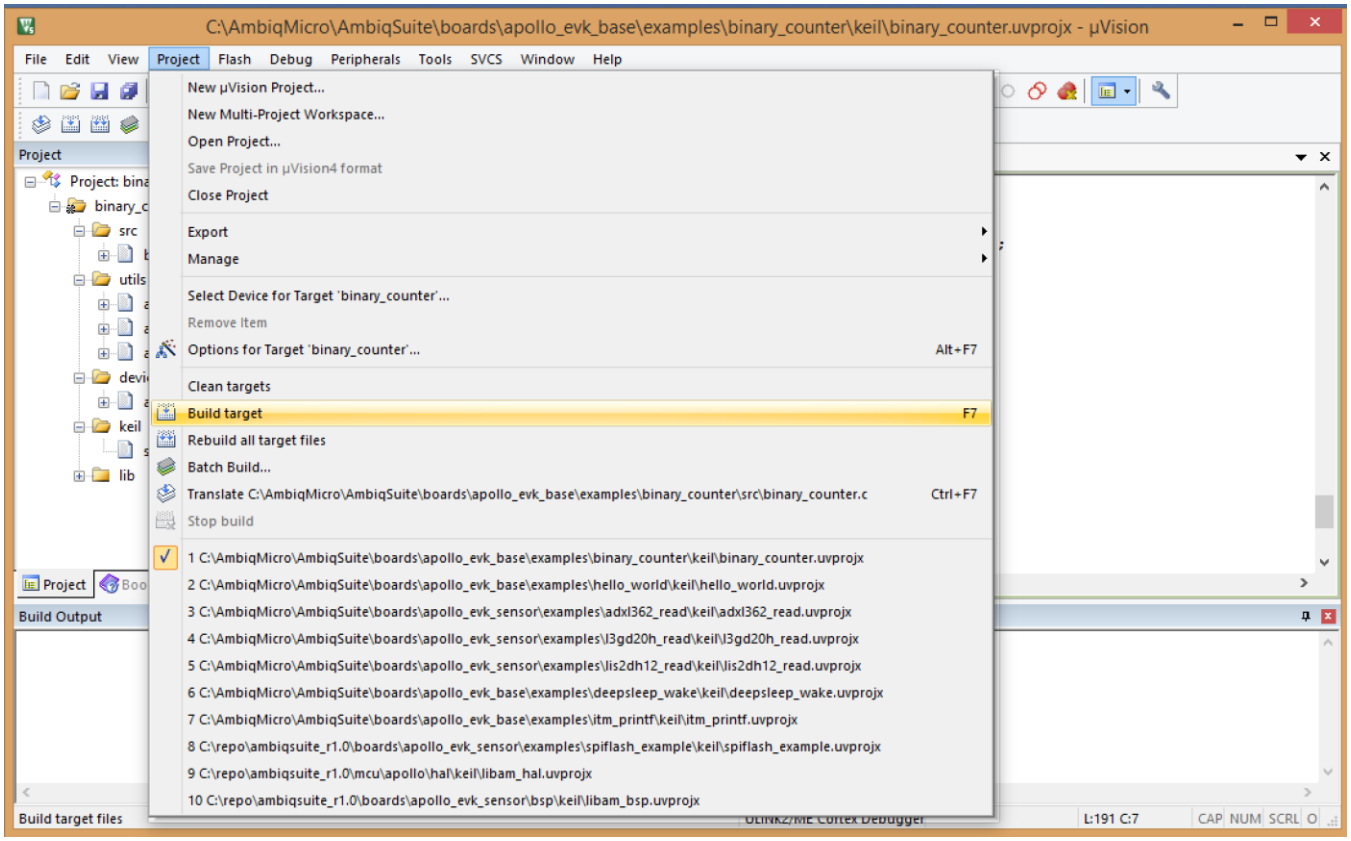


Figure 43 Build Keil Project for example "binary_counter"

4.3 Load program to flash

Make sure you have successfully built the project and setup the debugger. Load the image to flash at Flash→Download as shown in Figure 44.

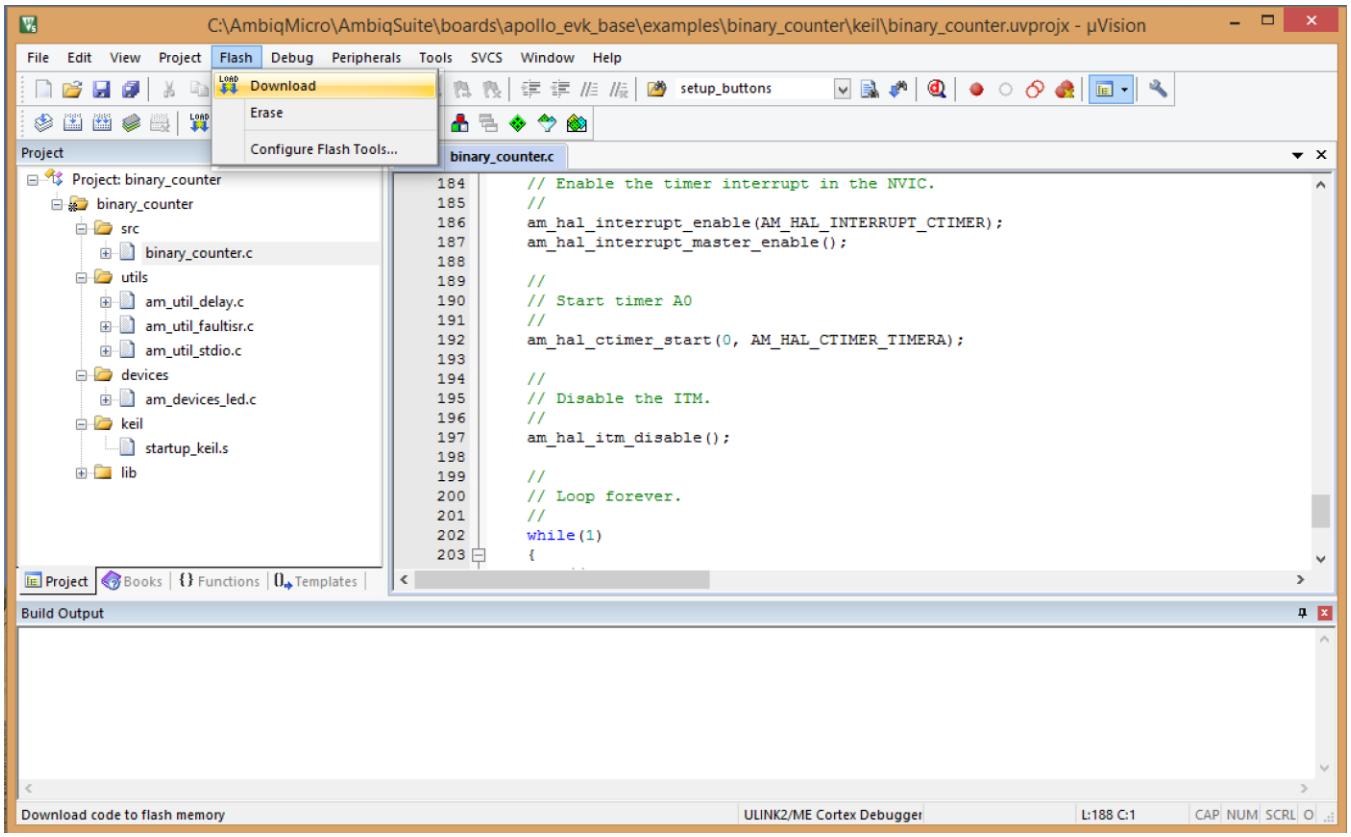


Figure 44: Load image to flash using Keil uVision

4.4 Debug

Make sure you have setup the debugger. Start debugging by clicking on Debug→Start/Stop Debugging as shown in Figure 45.

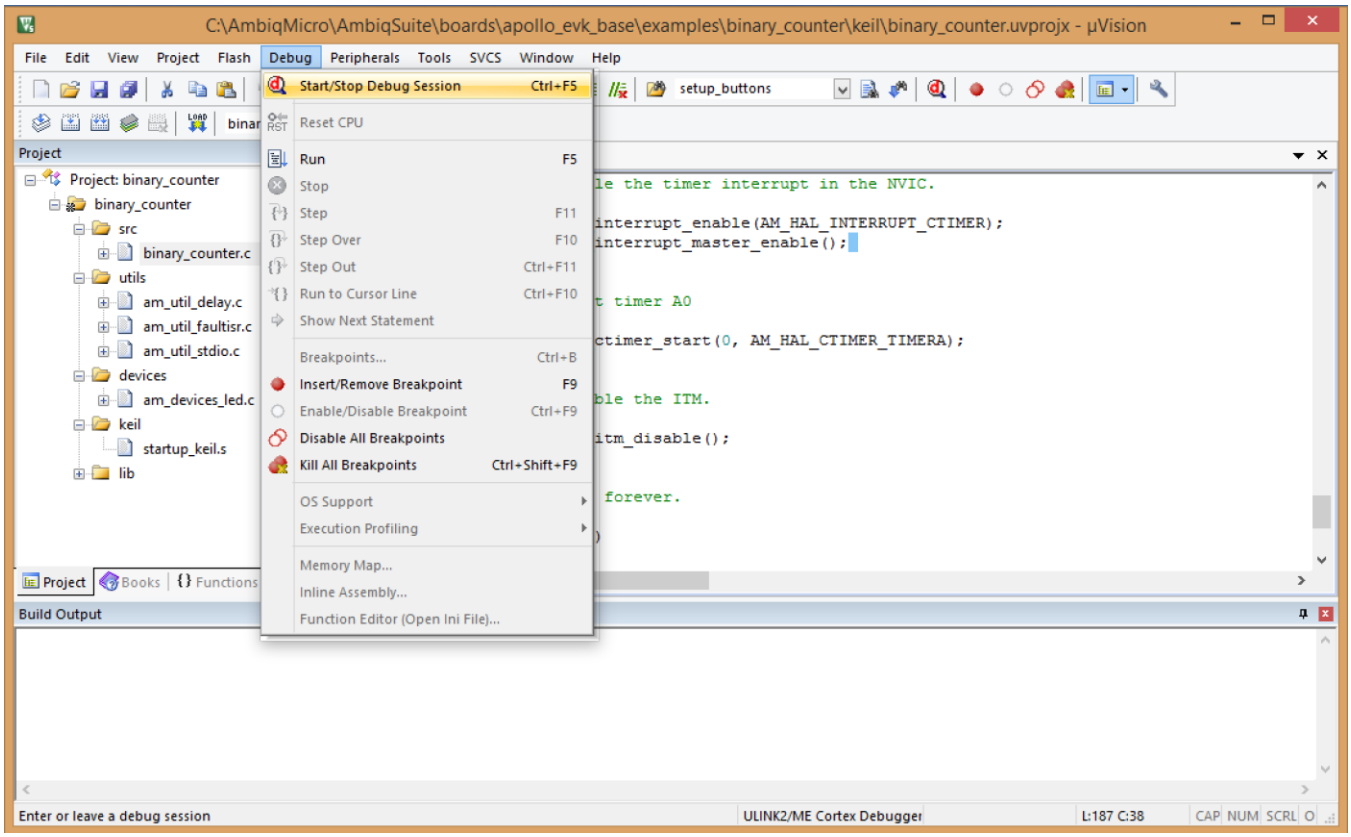


Figure 45: Start/Stop Debugging using Keil uVision

Once in Debug Mode open the Serial Debug Viewer Window to see debug printf messages from the Ambiq projects. Open this Window by clicking View→Serial Windows→Debug(printf) Viewer as shown in Figure 46.

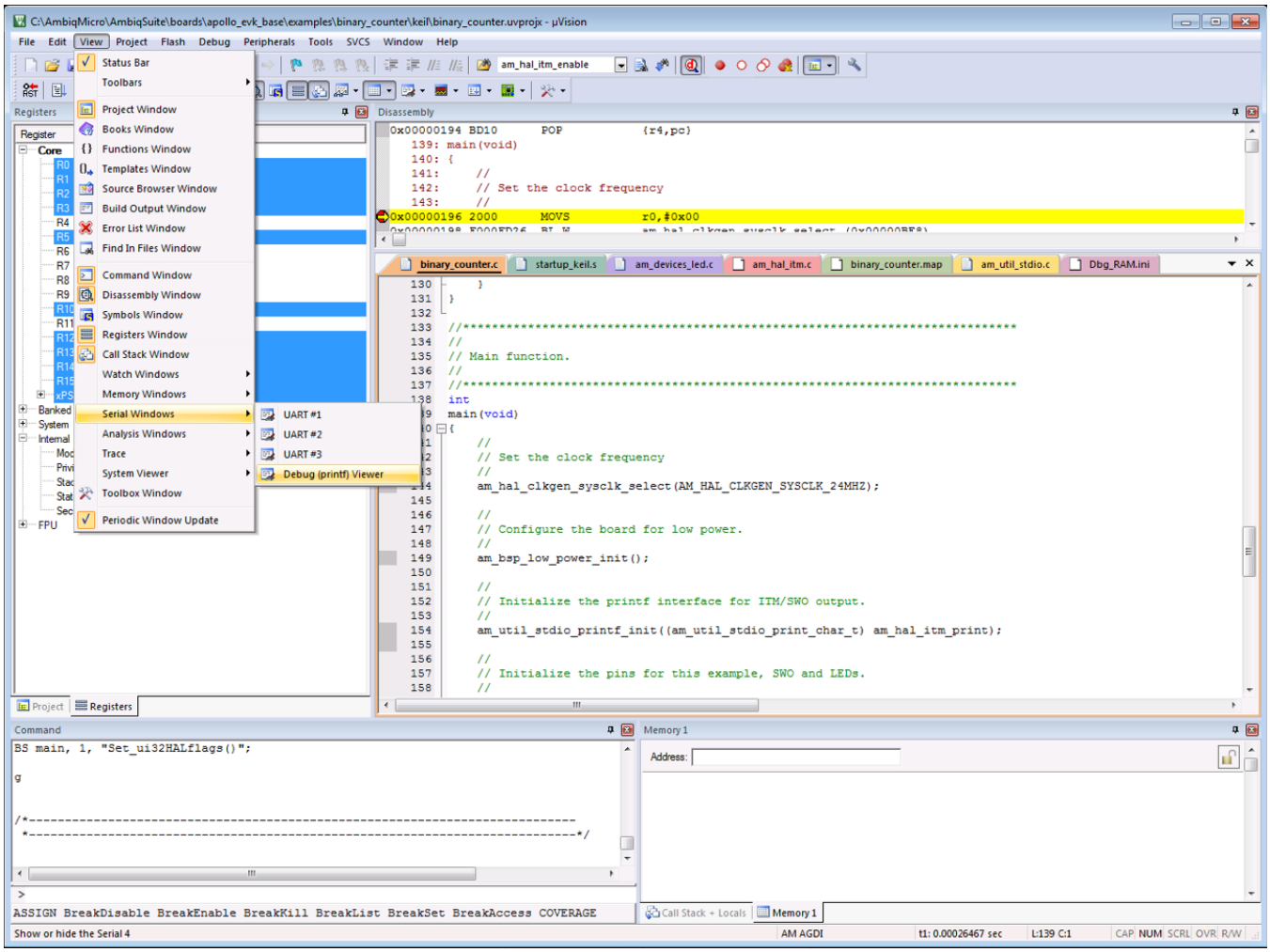


Figure 46: Open Serial Debug viewer

You should see the debug message from the example on the Debug(printf) Viewer as shown in Figure 47.

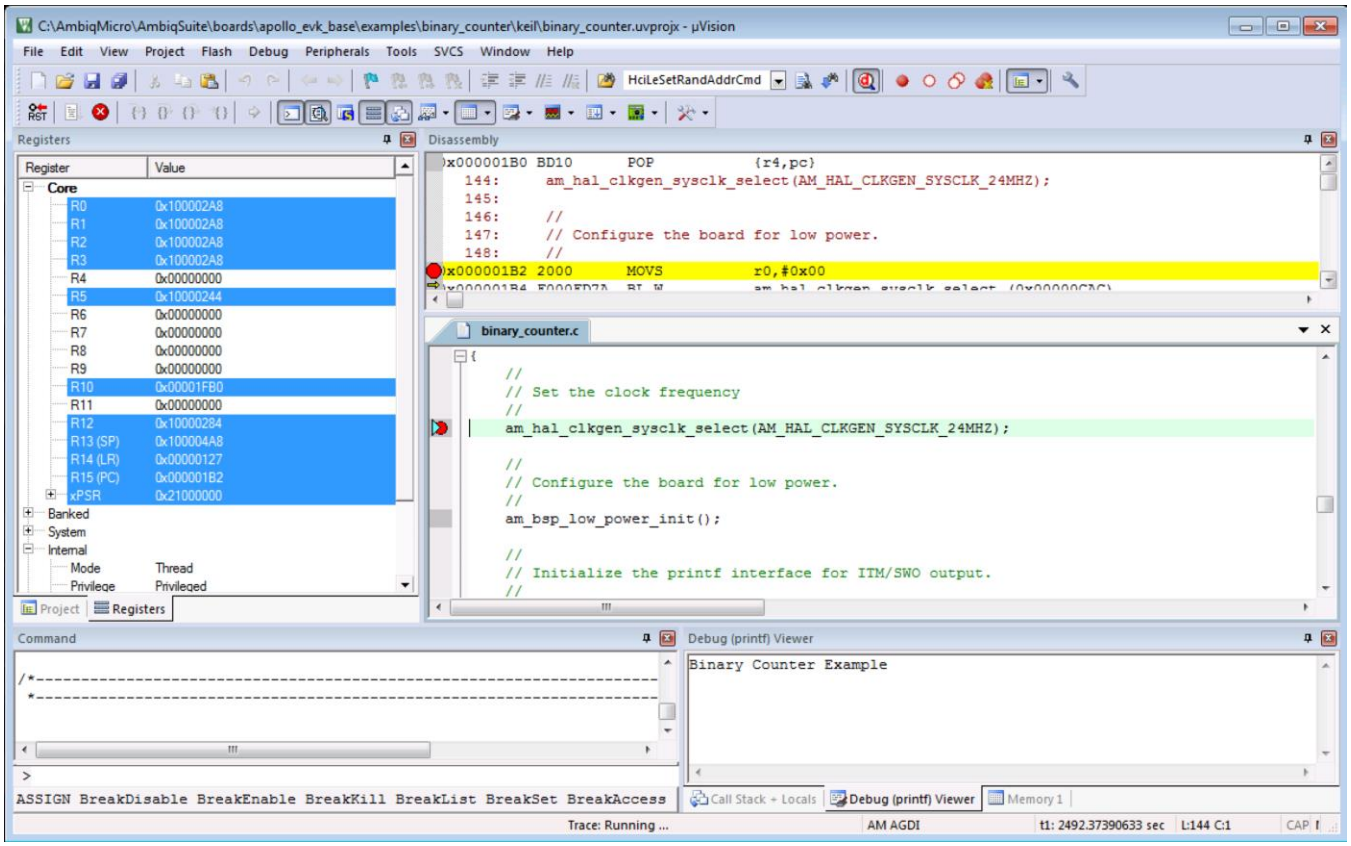


Figure 47: Debug (printf) Viewer

5. Build and Debug using IAR

Note: You should have IAR EWARM (minimum version v7.40.5) installed on the PC.

If you haven't already done so, go to the IAR EWARM download page at https://www.iar.com. Download the workbench and install it as directed by the documentation. If you will be using the a third party debug probe such as the IAR I-jet, then install it and its drivers now according to the manufacturers instructions. Connect the third party debugger to the Apollo EVK Base board and change the jumpers as described in the Apollo EVK board user's guide. Refer to Section 5.1.2 on debugger setup for third party debuggers.

If you would like to use the gdb server with the Ambiq Micro supplied Openocd driver for the EVK instead of third party debugger, then refer to section 5.1.1.

5.1 Debugger Setup

5.1.1 On-Board Debugger Setup

All Ambiq EVK boards ship with an on-board debugger that can be used with IAR. All Ambiq examples are configured by default to use the on-board debugger. Verify that the libusbK drivers are installed on your PC to use

the on-board debugger. To do this open Ambiq Control Center and check the ‘Drivers’ tab as shown in Figure 48. If libusbK driver is installed you can skip to Section 5.2.

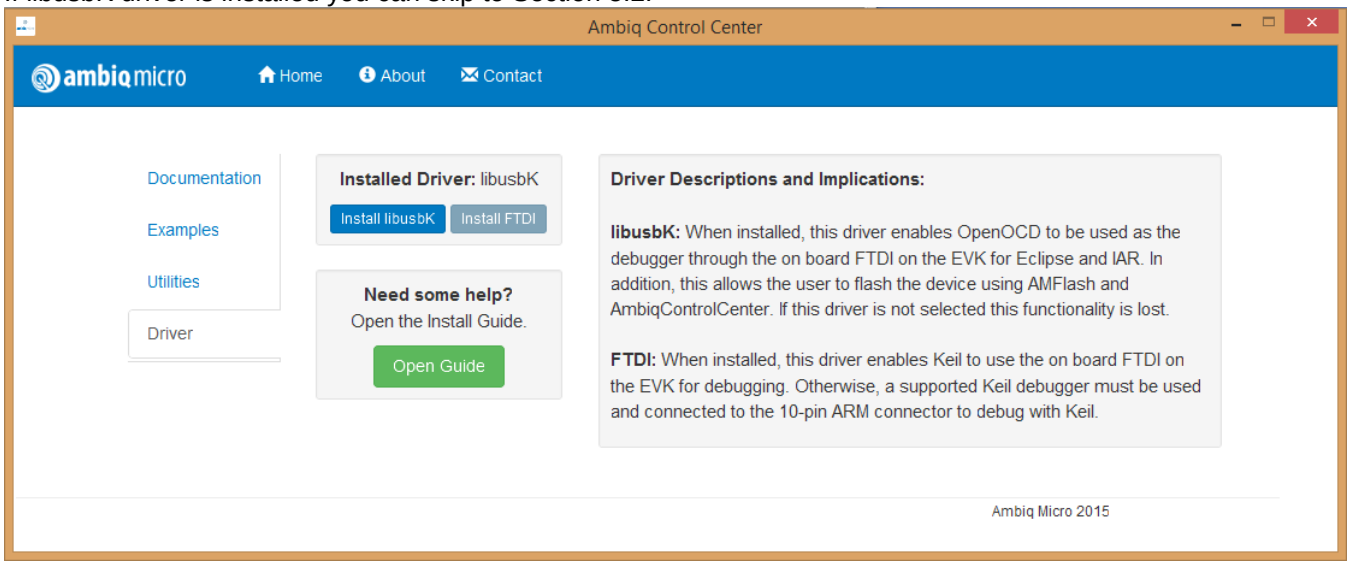


Figure 48. Verify that libusbK is installed in the ‘Driver’ tab

If the installed driver is “FTDI” then click on “Install libusbK” to install the libusbK driver. You will be prompted to have the EVK connected during installation. Please make sure it is connected and powered on during installation. You will be prompted one more time to verify that your EVK is connected and powered on – in case you missed it the first time! Double check and then continue with the driver install.

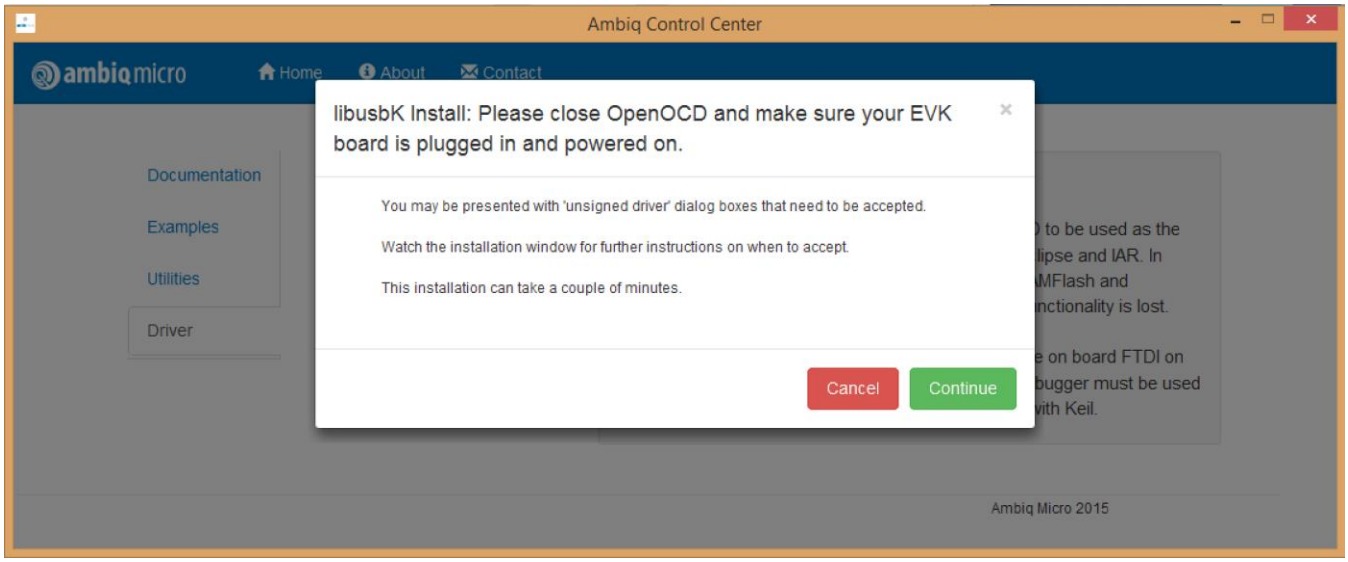


Figure 49. libusbK install: EVK check

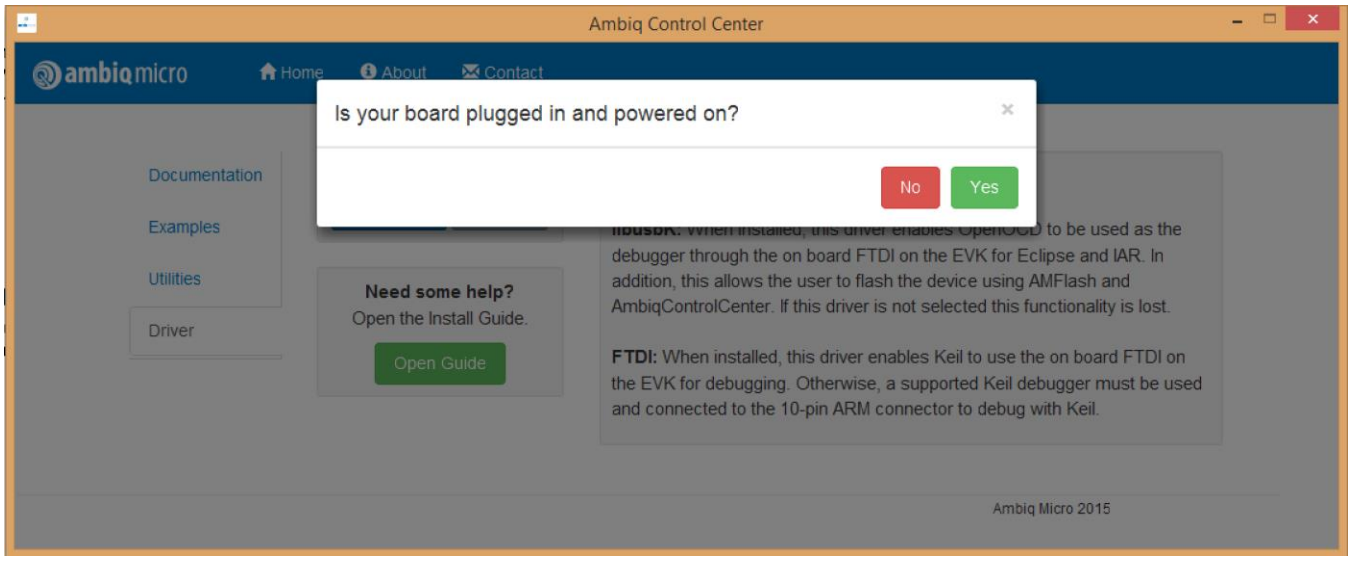


Figure 50. libusbK install: EVK double check

You will see a few windows pop up like the User Account Control window shown in Figure 51 and the Windows Security Alert window shown in Figure 52. Accept these to proceed with the installation. This could take several minutes.

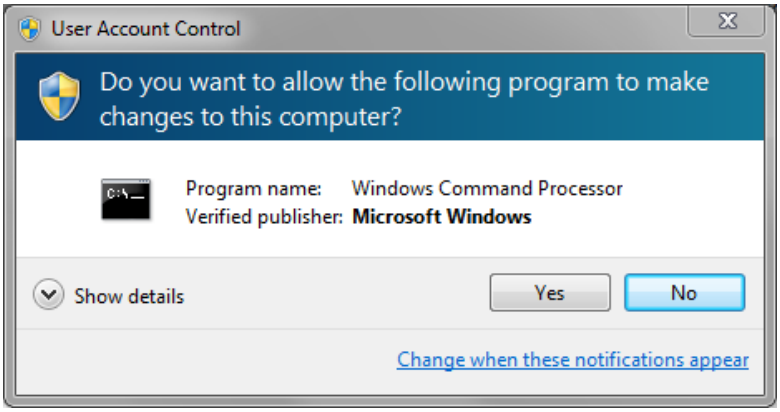


Figure 51. User Account Control Window

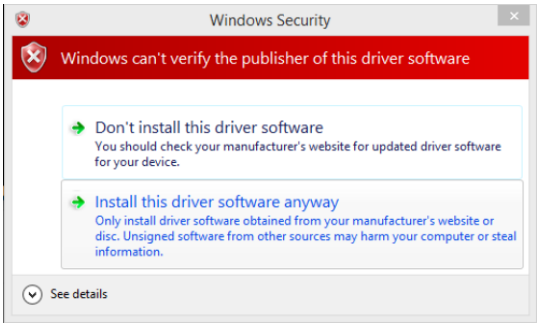


Figure 52. Windows Security Alert

At the end of the installation you should see the installed driver as “libusbK” as shown in Figure 48. Power cycle the EVK and start using the libusbK driver.

5.1.2 Third Party Debugger Setup (I-Jet)

To use a third party debugger verify that you have the drivers for the debug probe installed. Right click on the example workspace as shown in Figure 53 to option the ‘Options’ screen.

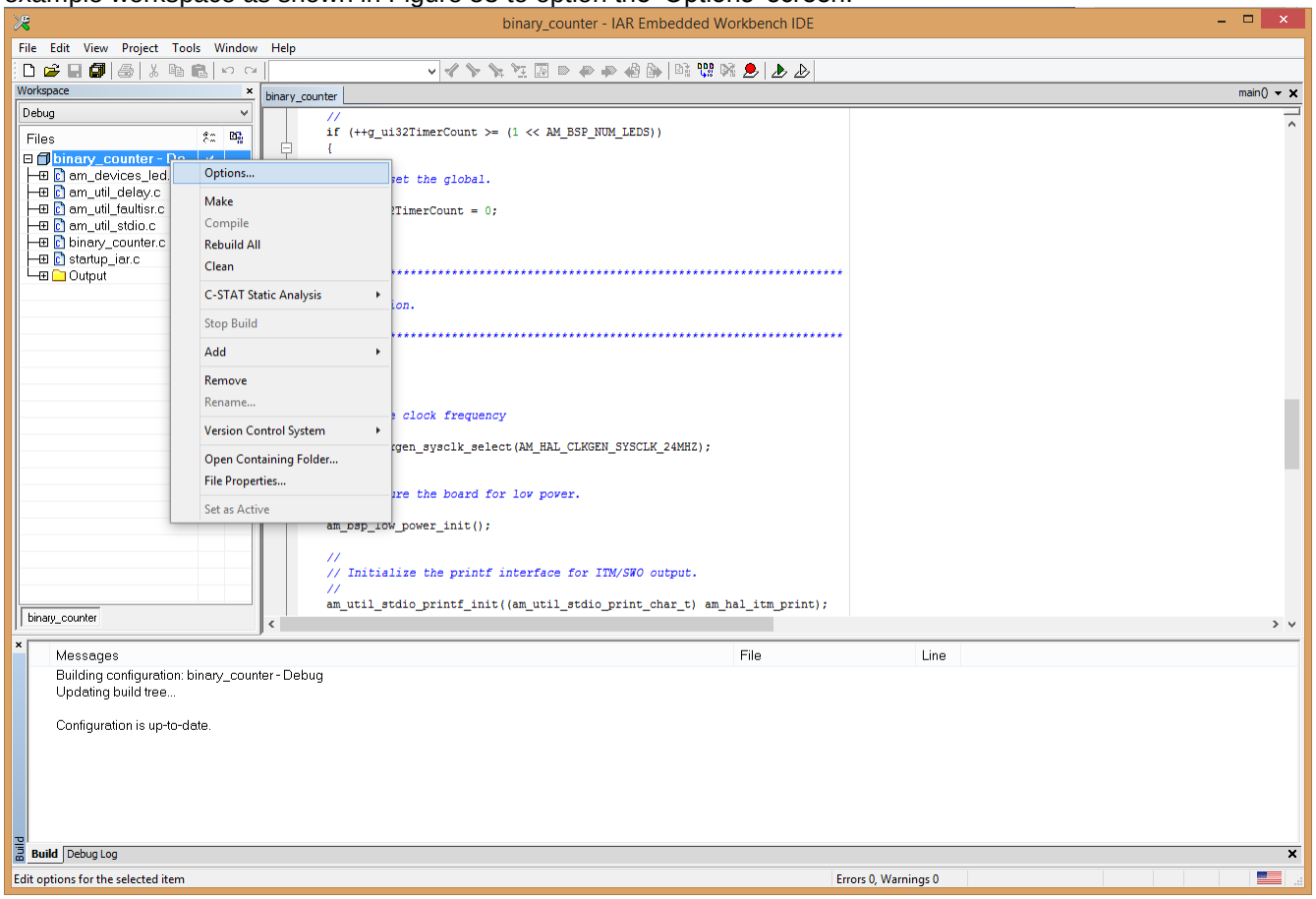


Figure 53. ‘Options’ screen for the workspace “binary_counter”

In the ‘Options’ screen select ‘Debugger’ in ‘Category’ and then select I-jet/JTAGjet (or your third part debugger of preference) from the ‘Driver’ drop down menu as shown in Figure 54.

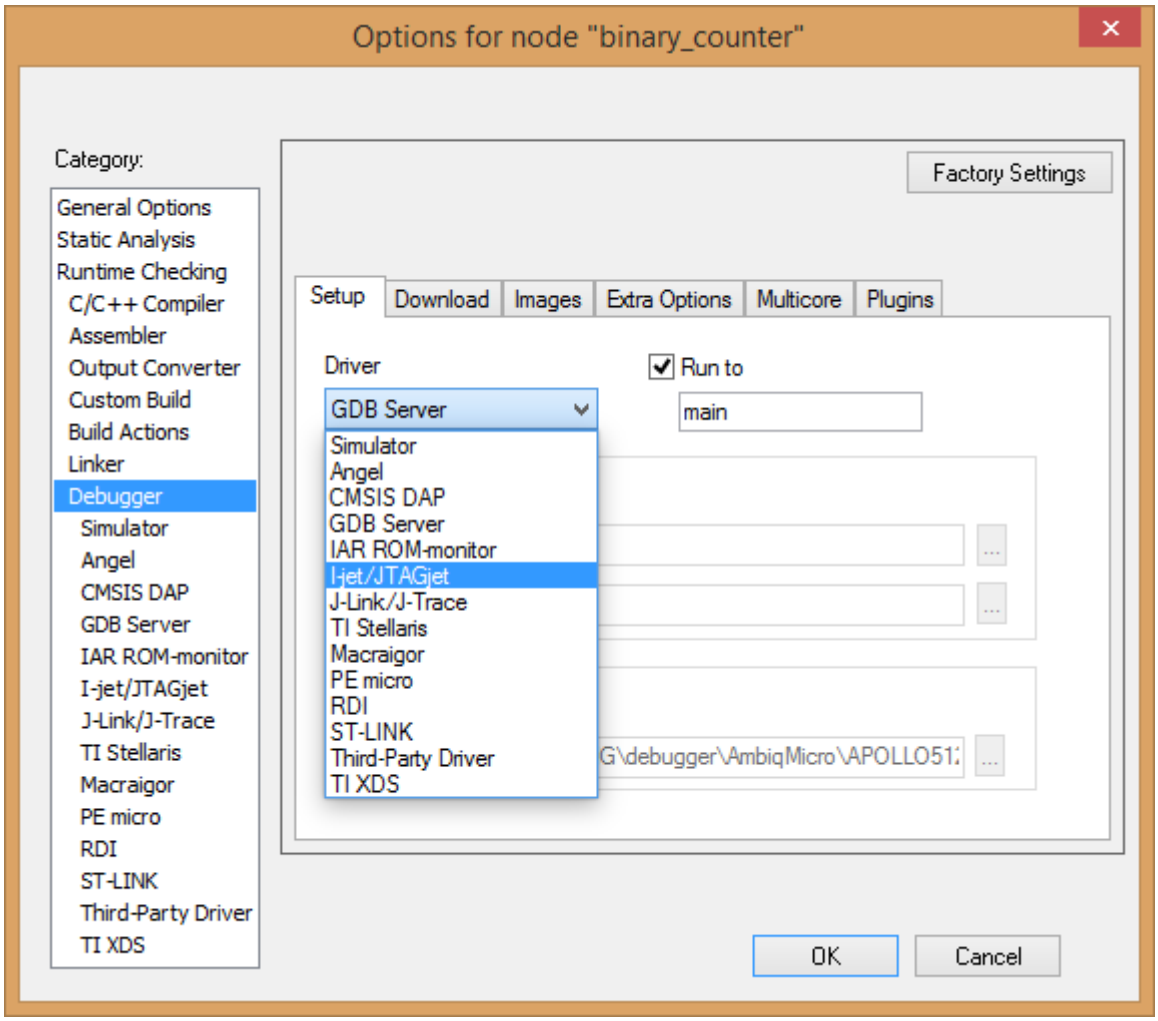


Figure 54. Third part debugger setup

5.2 Open a Project and Build

- Open an existing IAR project “binary_counter” from Ambiqsuite.

  The default location for Ambiqsuite project “binary_counter” is”:

  “C:\AmbiqMicro\AmbiqSuite\boards\apollo_evk_base\examples\binary_counter\iar”
- Double click on the IAR IDE Workspace file as shown in the figure 55 to open this projects using IAR.

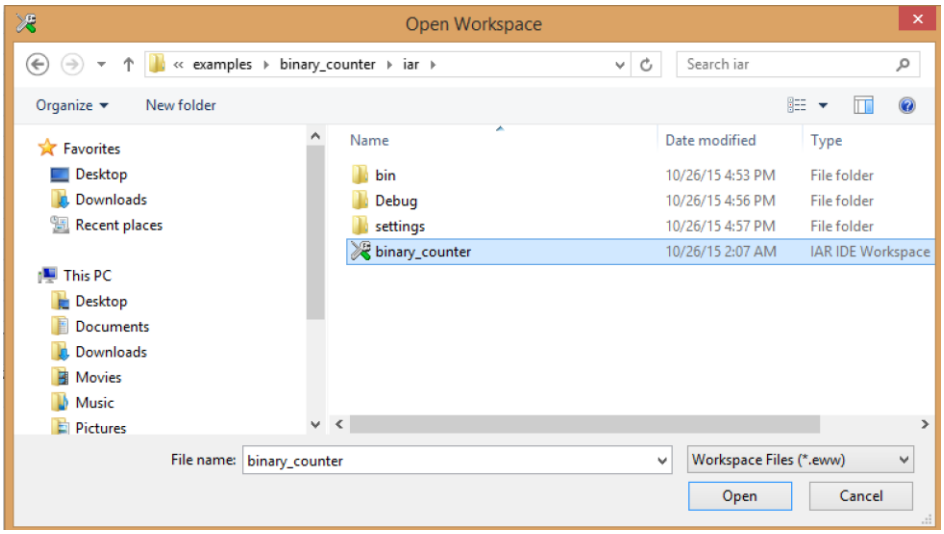


Figure 55: IAR Project folder for example 'binary_counter'

- Build the projects at Project→Make as shown in the figure 56.

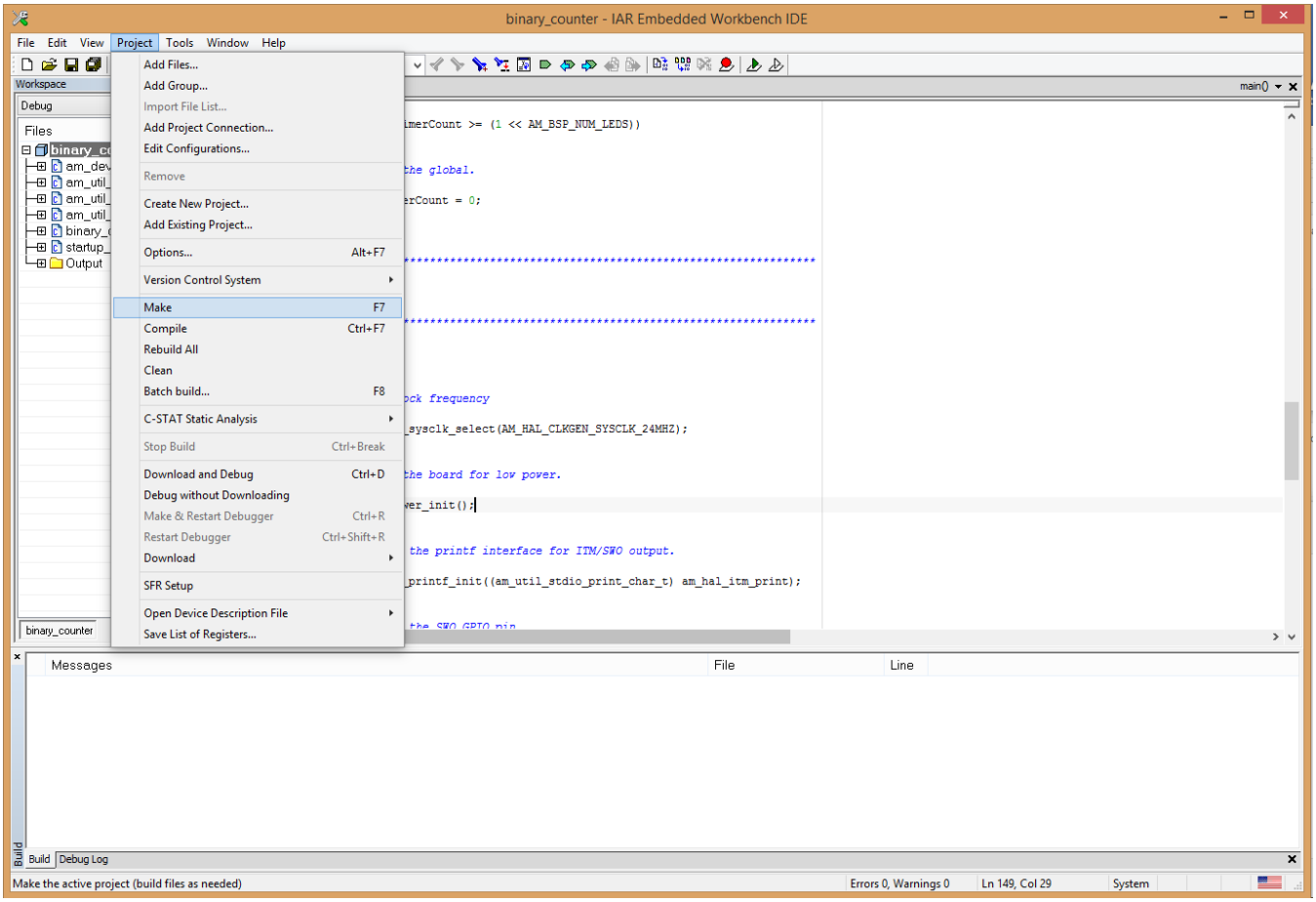
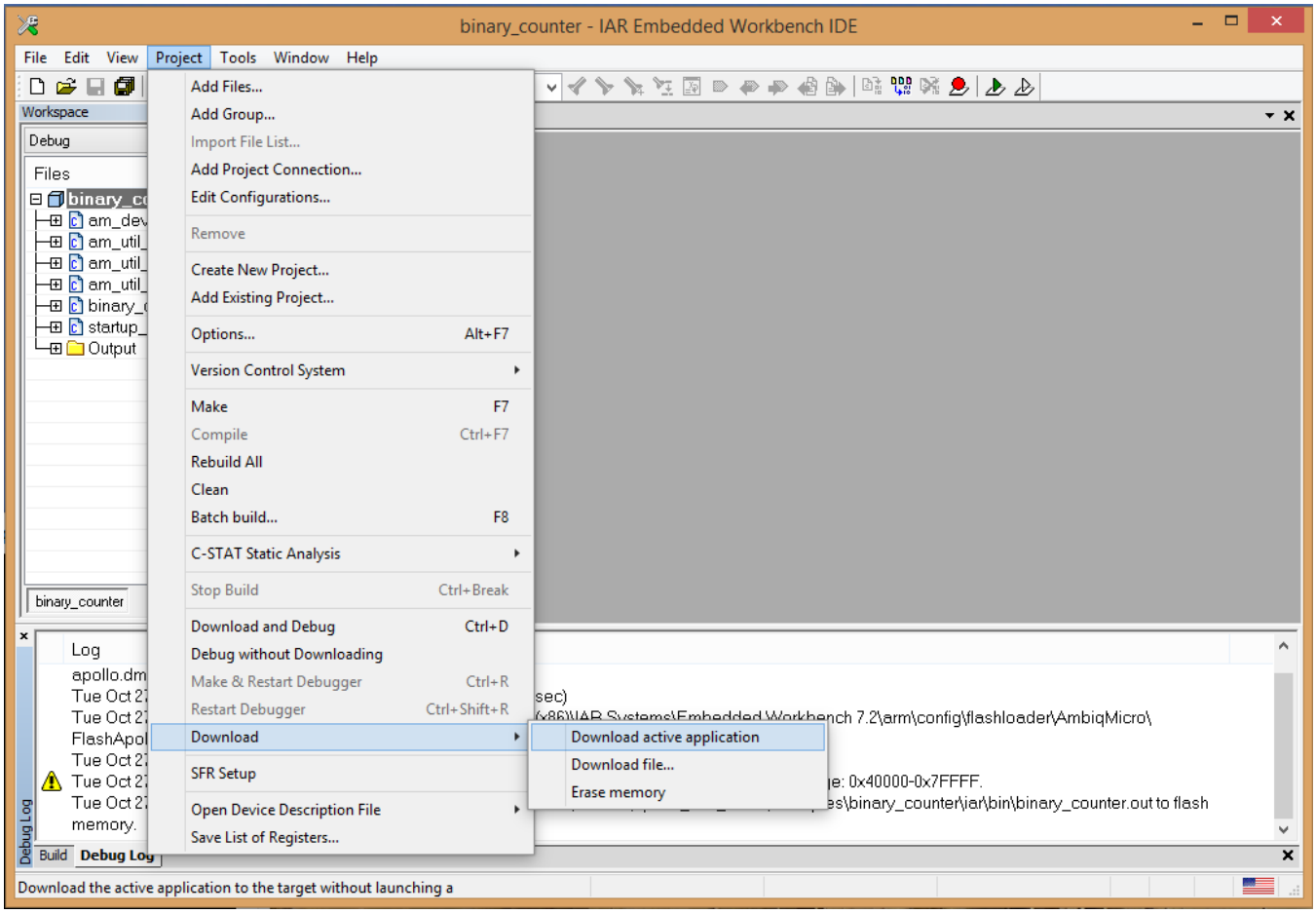


Figure 56. Build IAR Project for example "binary_counter"

5.3 Load program to flash

Make sure you have successfully built the project and setup the debugger. Start OpenOCD. Refer to Section 3.2.2 for a refresher.
Load the image to flash at Project→Download→Download Active Application as shown in Figure 57.



.

Figure 57. Download to Flash

You will very briefly see a progress bar like the one in Figure 58 while the image is written to flash.

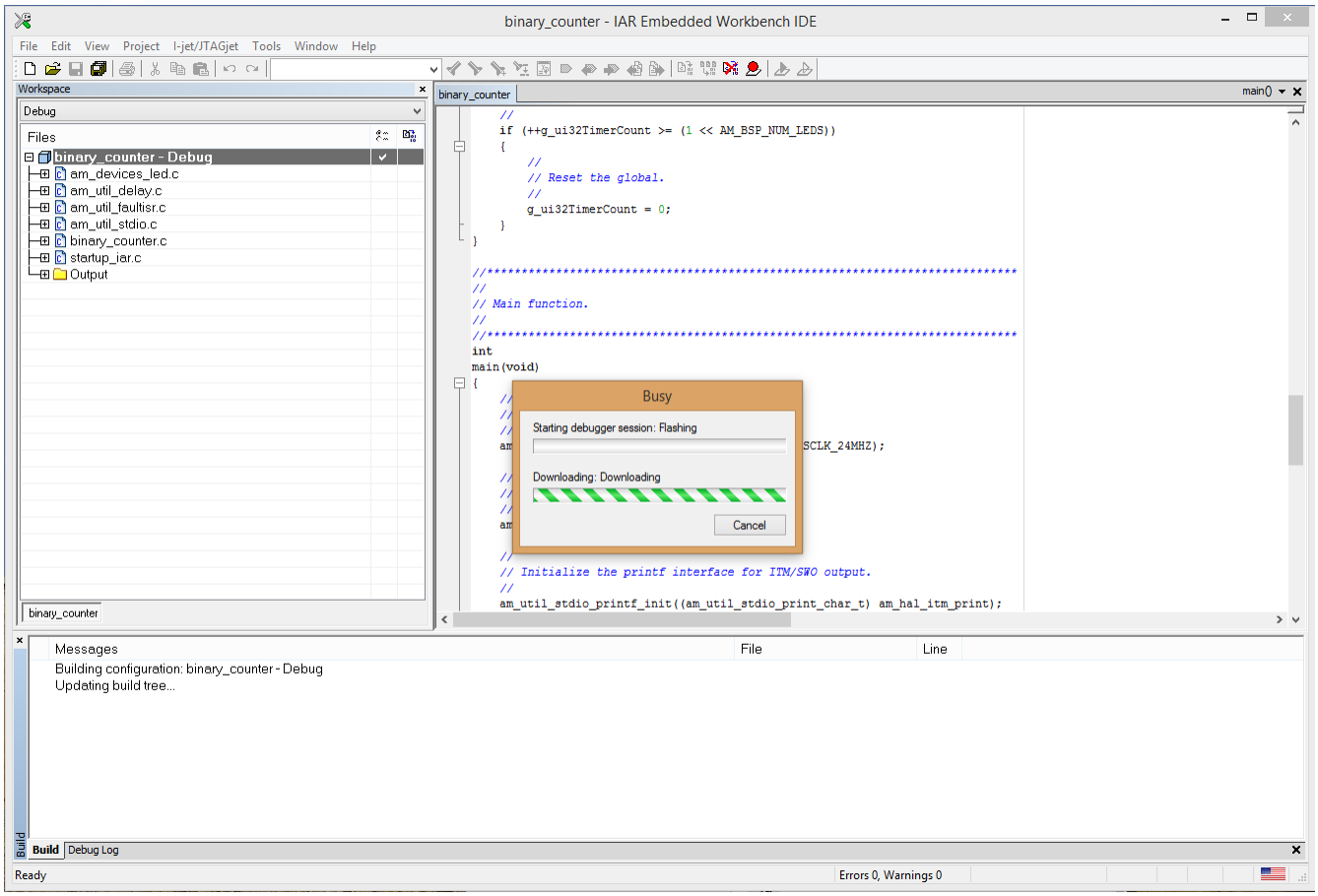


Figure 58. Download to Flash in progress

5.4 Debug

Make sure you have setup the debugger. Start debugging by clicking on Project→Download and Debug as shown in figure 59.

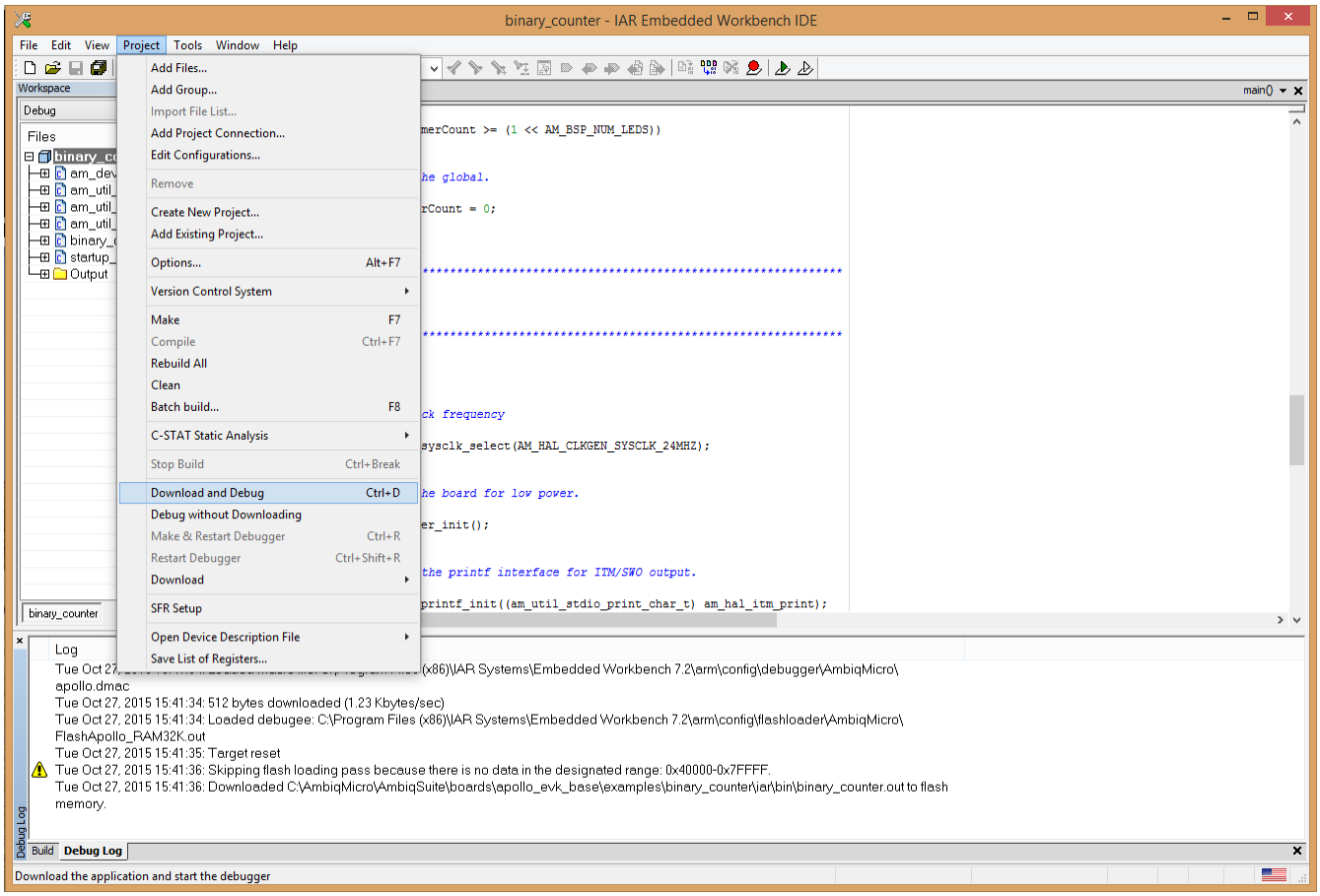


Figure 59. Download and Debug

6. Troubleshooting

There are several situations that can cause connectivity issues between the PC and the Apollo device. This section explains some common issues and their solutions.

6.1 Physical Debugger Connection Issues

One of the most common reasons for connectivity issues between the PC and an Apollo EVK is related to jumper settings. The Apollo EVK has multiple jumpers related to power and SWD signal routing. If any of these have been placed incorrectly, OpenOCD will either report the absence of a device, or it may report a device with the wrong number of breakpoints and watchpoints. Also, all tools that access the Apollo MCU over SWD (including Eclipse, AM Flash, and Ambiq Control Center) will report errors. Setting the power and SWD jumpers back to their default positions (which can be found in the User's Guide for the EVK) will solve this issue.

6.2 Windows Driver Issues

If an Apollo board is unresponsive, even with all jumpers in their default positions, it's possible that the connection issue is actually related to a Windows USB Driver problem. To check for USB driver issues, open the Windows Device Manager. This can be found under the Control Panel -> Hardware and Sound.

For a correctly enumerated Apollo EVK, the device manager should have entries that look like the following images:

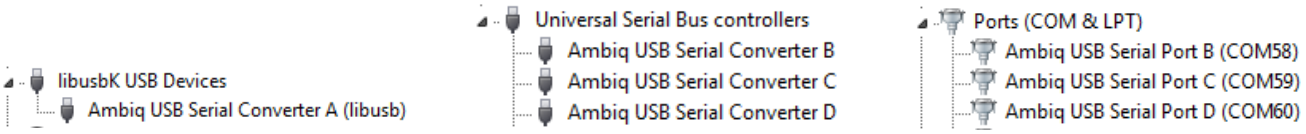


Figure 60: Device Manager Entries

If any these entries are missing or incorrect, some or all of the SWD or ITM/SWO connections to the board will be unusable. To resolve this issue, you will need to re-install the device drivers for the Ambiq debugger hardware. This can either be done automatically by running the Ambiq Driver Installer executable, or, if Ambiq Control Center is already installed it can be performed manually.

To re-install drivers manually, first open the Windows Device Manager and locate the devices that have missing or incorrect drivers. Usually, they devices will appear in the device manager in the following form:

Figure 61: Missing Drivers

Right-click on one of the devices, and select "Update Driver Software". This will open the following dialog. Select "Browse my computer for driver software", and provide the path to the AmbiqDrivers directory that installs along with Ambiq Control Center. The default location for this folder is "C:\AmbiqMicro\AmbiqDrivers". Make sure that the "Include subfolders" option is selected, and click "Next".

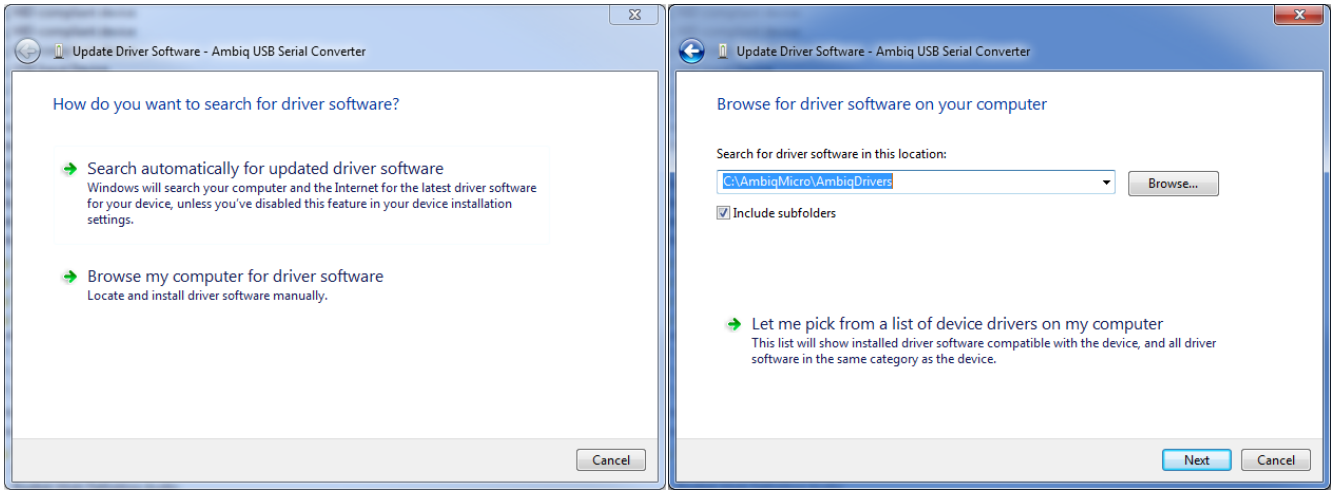


Figure 62: Updating Drivers

Windows may report that the drivers are unsigned or that the manufacturer cannot be identified. This is normal, especially for early versions of the Ambiq debugger-interface drivers. To continue with driver installation, select “Install this driver software anyway”.

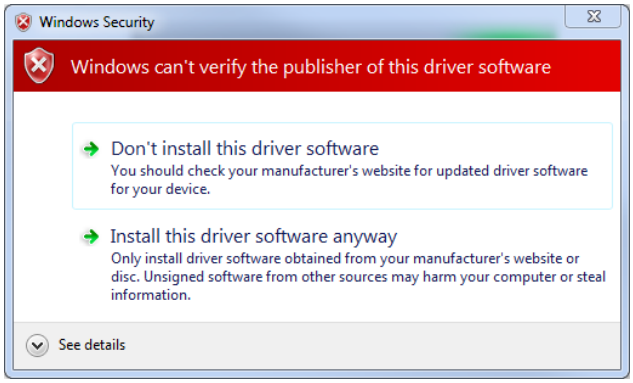


Figure 63: Driver Verification

If the installation is successful, you should see the following window appear. You will need to repeat this process for each of the USB interfaces with missing drivers. After this is complete, the Ambiq debugger tools should be able to connect to the Apollo EVK correctly.

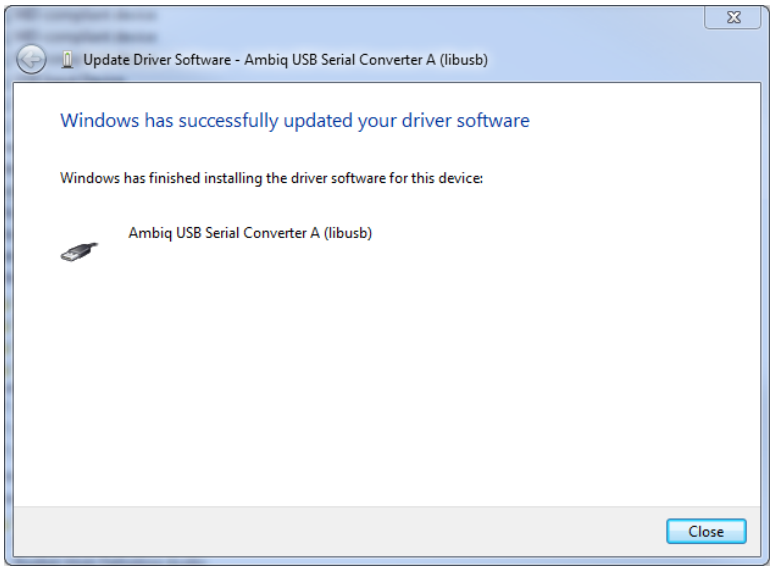


Figure 64: Successful Driver Installation

Document Revision History

Rev #	Description
0.01	Draft
0.02	Adding info about 3 tool chains (gcc,Keil,IAR).
0.03	Adding minimum IDE versions with Ambiq Supprot.

Contact Information

Address:

Ambiq Micro, Inc.
6500 River Place Blvd.
Building 7, Suite 200
Austin, TX 78730

Website:

www.ambiqmicro.com

General Information:

info@ambiqmicro.com

Sales:

sales@ambiqmicro.com