PAC Firmware Development Getting Started Guide

Power Application Controller®

2019-08-27

Table of Contents

1 Introduction

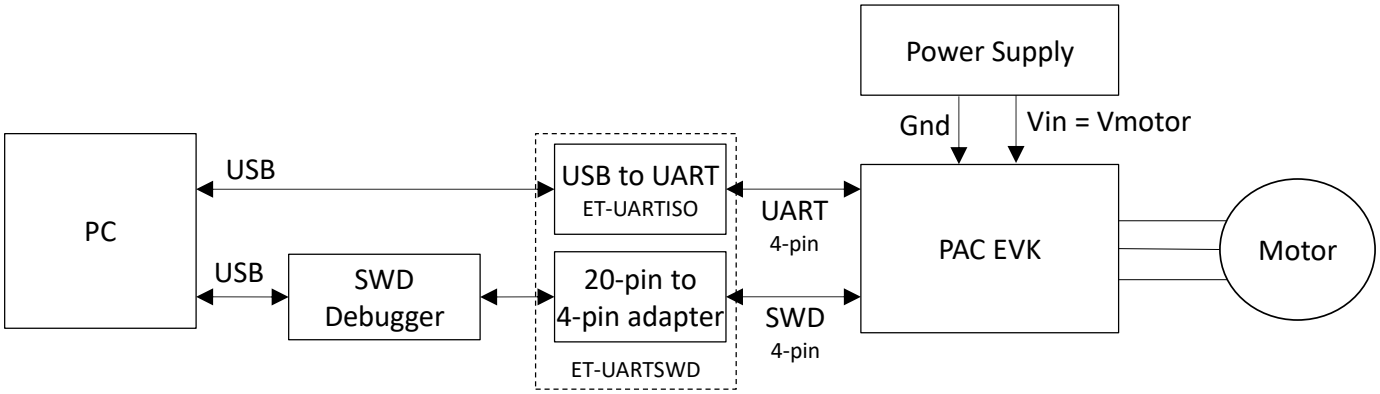
The PAC Firmware Development Getting Started Guide is a generic guide for the majority of PAC firmware releases and is designed to help get PAC firmware up and running on a PAC EVK. Because this guide is generic, other documents that are part of the firmware release should be consulted including any readme.txt files. This guide assumes that the desired IDE has been installed and if using a PAC55xx, the appropriate PAC55xx IDE support files have been added.

2 Development Setup

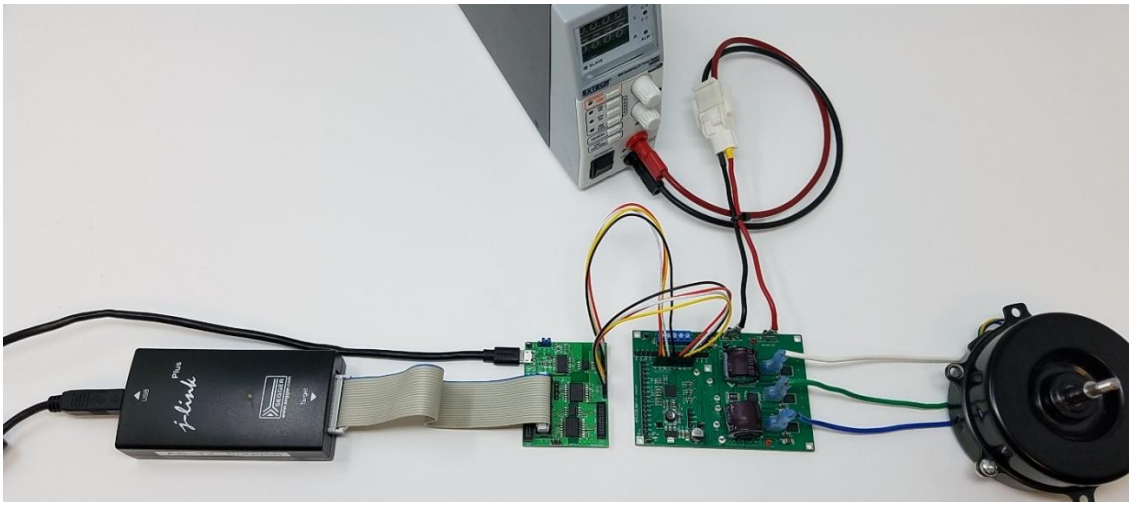
The first step is to setup the development environment as shown in the figure below; for some PAC example firmware the motor is optional. The setup must include the following:

1. PAC EVK board
    a. These are normally shipped with the FLASH erased
2. Power Supply
    a. Should be capable of supplying Vin rated for the particular PAC EVK and/or optional motor
3. SWD debugger
4. Adapter Board(s) for connecting the SWD debugger and UART
5. Windows PC with desired IDE.

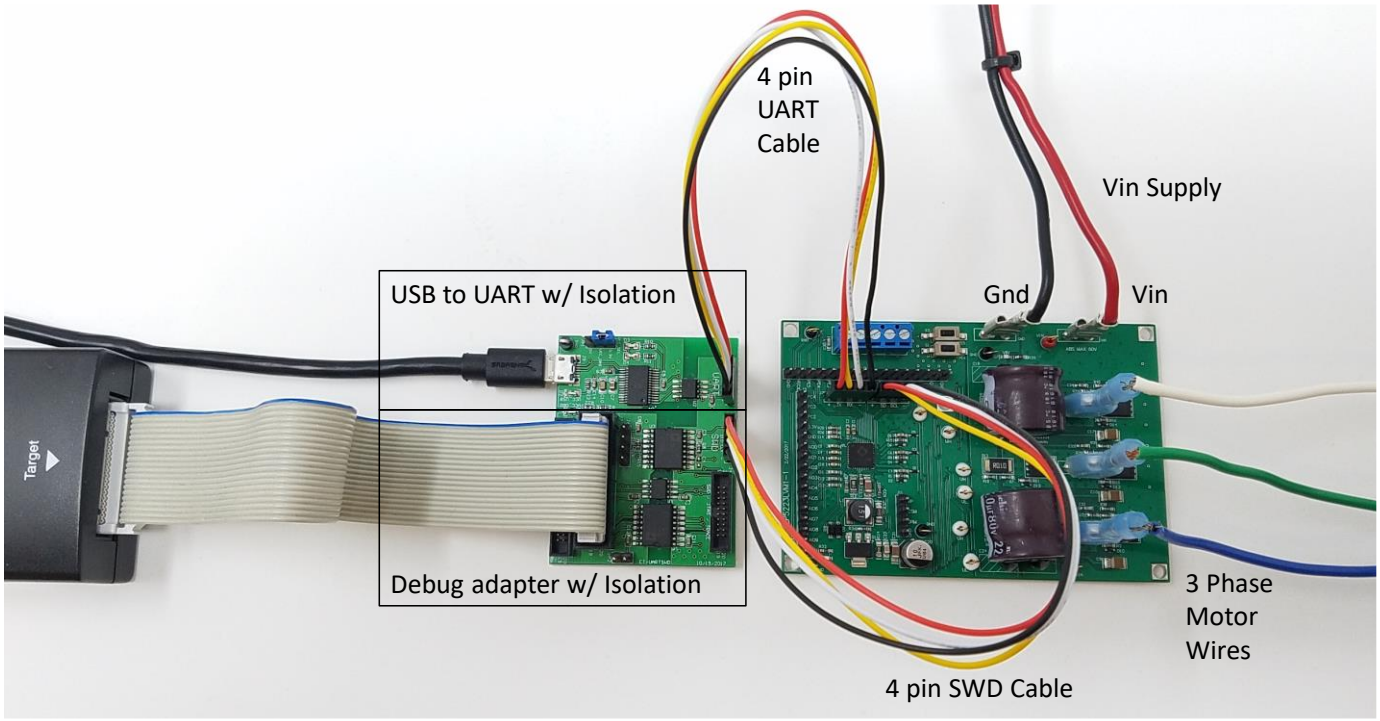
The figure below illustrates the different components.



The picture below illustrates the setup for a PAC5223 EVK using the ET-UARTSWD adapter board.



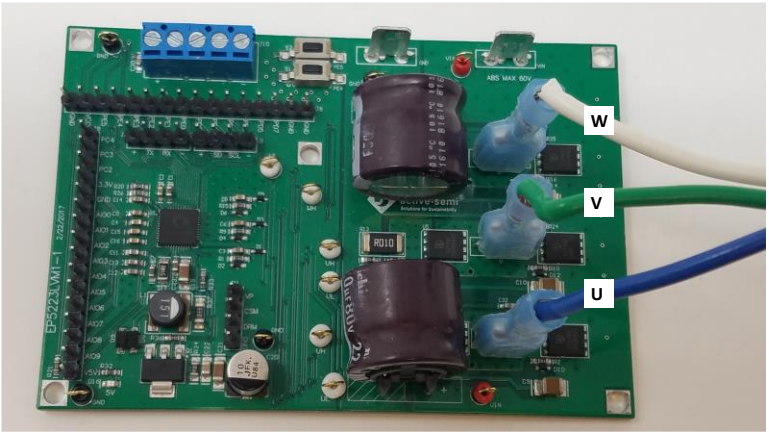
This picture is a close-up of the EVK connections for the PAC5223 EVK.



This getting started guide will walk through each connection step by step. Images in this guide only show the setup of the PAC5223 EVK, so please consult the PAC EVK user's guide for details of the specific PAC device being evaluated.

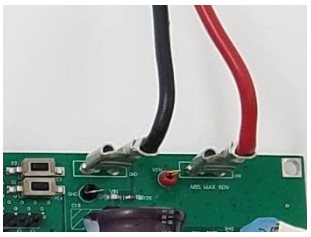
2.1 Motor Connection (optional)

For some PAC firmware examples, connecting a motor is not required. But, if evaluating the PAC BEMF/BLDC or FOC firmware, then connect a 3 Phase BLDC/PMSM motor using the spade connectors labeled U, V and W that are present on PAC EVKs. The image below shows the connection to the PAC5223 EVK.



2.2 Power Supply Connection

Connect a power supply to the Vin and Gnd input spade connectors on the PAC EVK according to the PAC EVK ratings, and/or the motor specification if used. The image on the right shows the connection on a PAC5223 EVK.



Wait to turn on the supply until just before flashing the device with an IDE.

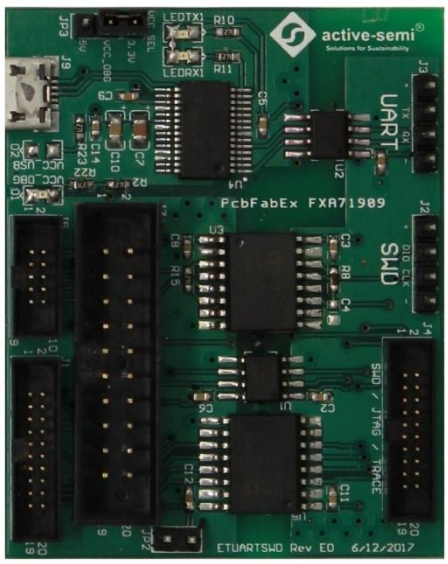
2.3 PC UART and Debugger Connection

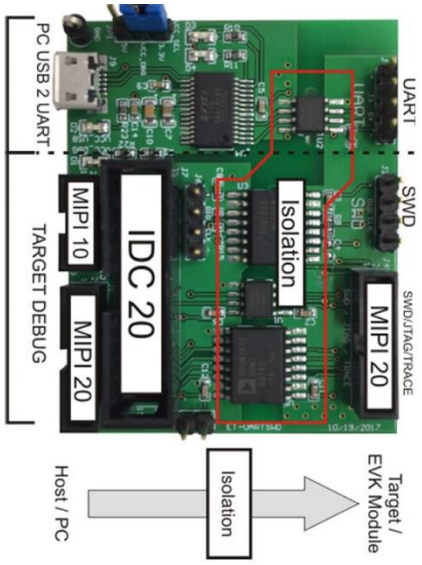
Some of the PAC firmware makes use of a UART for output to a Windows terminal or for control using a Windows based GUI. These require a UART connection between the PC and the PAC device. Also, a debugger is required to flash the firmware to the PAC device. Two primary options are available for the UART and debugger connectivity:

1) ET-UARTSWD combination adapter board
2) ET-UARTISO-1 with either ET-IARISO-1 or jumper SWD debugger connection

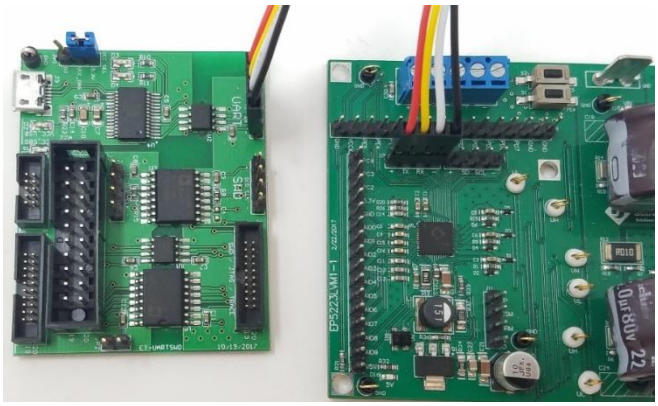
2.3.1 Option 1: ET-UARTSWD

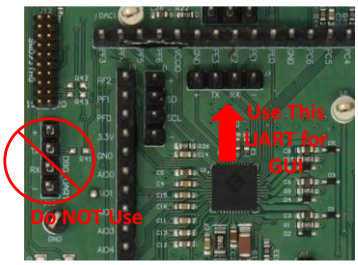
Using the ET-UARTSWD adapter board is the best method for UART and Debugger connectivity. New PAC EVKs are now shipping with the ET-UARTSWD combination adapter board that includes isolation for high voltage applications (see image below on left). The ET-UARTSWD has both a USB to UART function and connectors that adapt various debugger connector types to the 4-pin SWD header found on all PAC EVKs. It also includes an optional target side 20-pin MIPI connector for PAC55XX EVKs. Various aspects of the board are highlighted in the image below on the right.





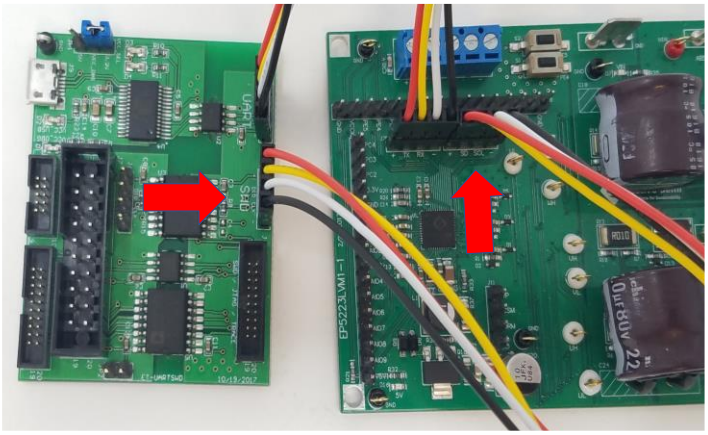
To connect the UART, use a 4 wire cable to connect the PAC EVK 4-pin UART header labelled [+ TX RX -] to the ET-UARTSWD 4-pin UART header labelled [+ TX RX -]. The image on the right shows the connections for the PAC5223 EVK.



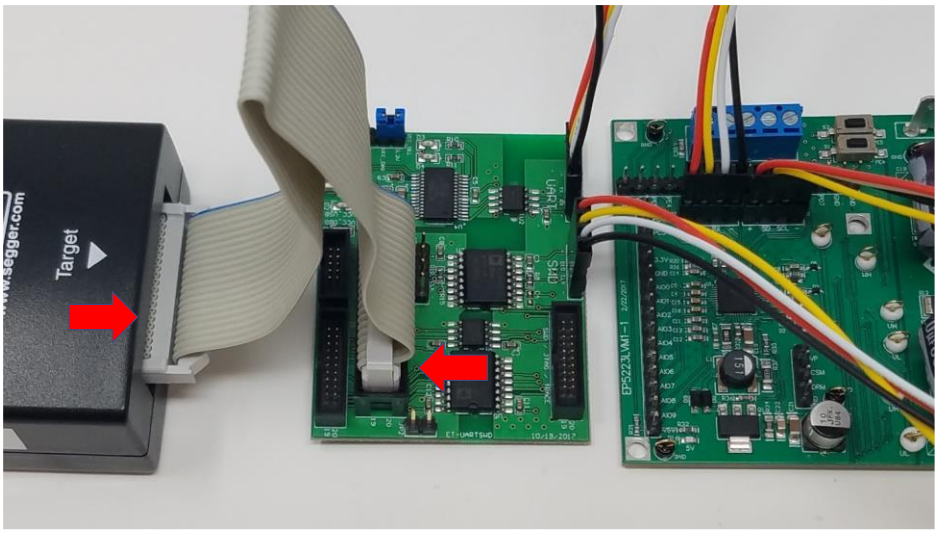


Note: On PAC55xx EVKs make sure the cable is not connected to the DBG_UART. The image shown on the left highlights the 2 UART headers on the PAC5523 EVK. The DBG_UART is primarily used for printfs() via UART to a PC terminal applications.

Next, use a 4 wire cable to connect the PAC EVK 4-pin SWD header labelled [+ SD SCL -] to the ET-UARTSWD 4-pin UART header labelled [+ DIO CLK -]. The image below shows the connection for the PAC5223 EVK.



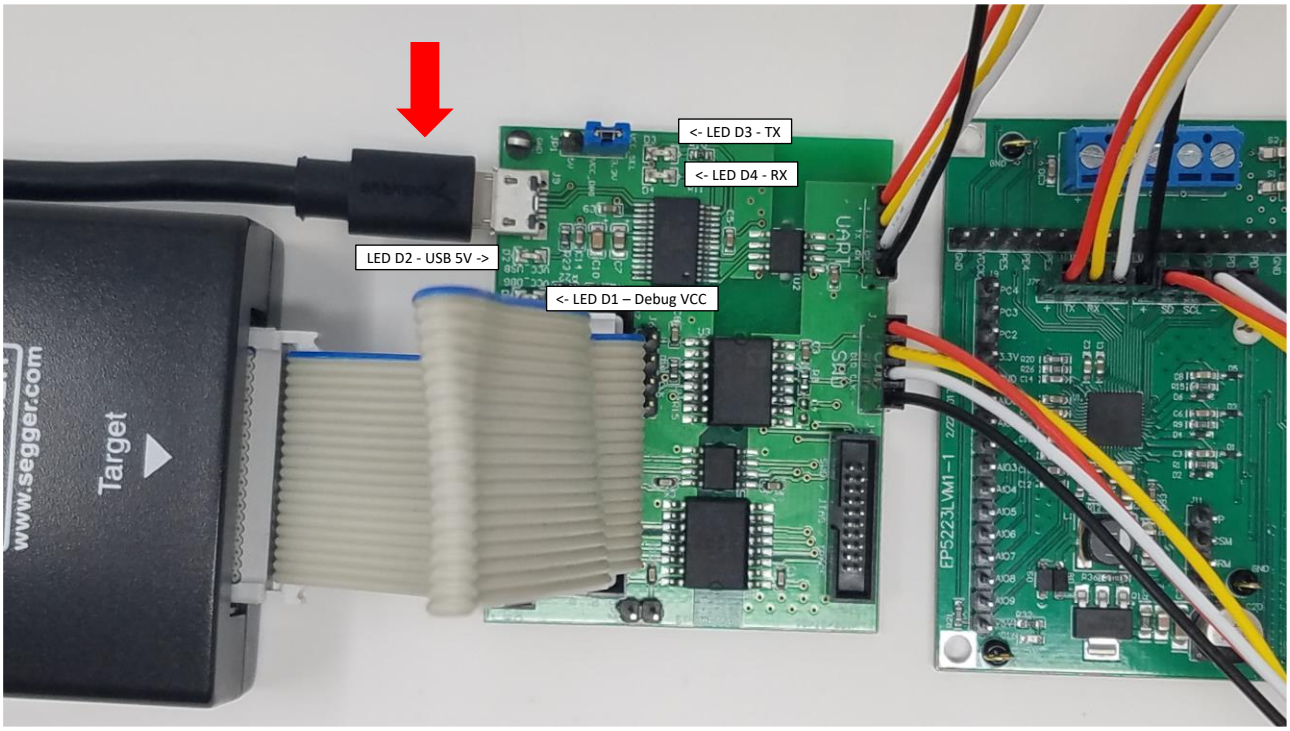
Now connect the 20-pin IDC ribbon cable of the SWD debugger to the appropriate PC/Host side connector on the ET-UARTSWD board. The image below shows a J-Link 20 pin IDC ribbon cable connection.



Notes:

1) The majority of debuggers sense the target hardware Vcc on pin 1 and can handle a 3.3v input. So, the most typical configuration for the ET-UARTSWD is where the power for the PC/Host side of the isolation is supplied by the USB to UART circuitry. For this configuration, the ET-UARTSWD jumper JP3 should be connecting 3.3v to VCC_DBG, which is the default position. If the debugger can't support this because
   a. it supplies a voltage on pin 1,
   b. it doesn't support 3.3v input, or
   c. if a USB connection to the PC is not being used, then consult the ET-UARTSWD User's Guide for powering the debugger Vcc.
2) Debuggers that have built in isolation are not supported, because the 2 isolation circuits will interfere with each other.
   a. The ET-COLINK-1 debugger for CooCox includes isolation and so is not supported with ET-UARTSWD. ET-COLINK-1 can be directly connected to the 4-pin SWD header.
3) For debuggers with built in isolation, connect the 4 pins [Vcc, SD, SC, Gnd] directly to the SWD 4 pin header on the PAC EVK using jumpers.

Next connect a USB cable from the PC to the ET-UARTSWD adapter board as illustrated in the image below. If USB power is present, then LED D2 should light up. LED D1 should also light up indicating the debug Vcc is active. D3 and D4 LEDs will turn on and off based on UART TX and RX activity respectfully.

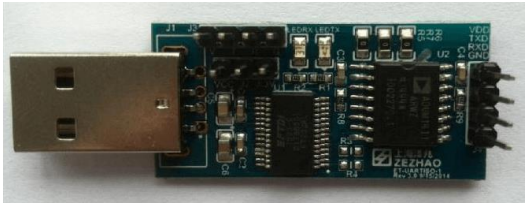


Finally, plug the debugger's USB cable into the PC.

Now proceed to paragraph 3 to build and flash the PAC firmware image.

2.3.2 Option 2: ET-UARTISO-1 with ET-IARISO-1 or jumper SWD connection

Another option for UART connectivity is to use the ET-UARTISO-1 adapter for USB to UART communications. In this case, SWD connectivity to the PAC EVK is handled separately. The ET-UARTISO-1 is shown below.

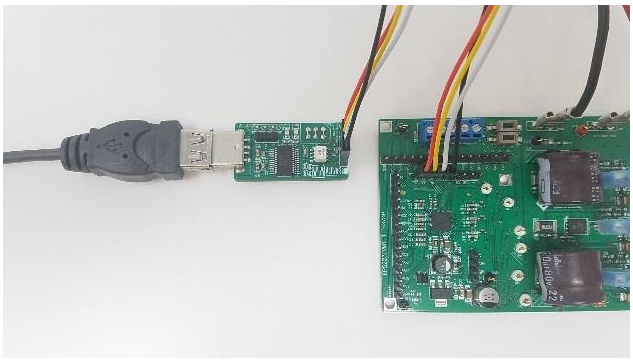


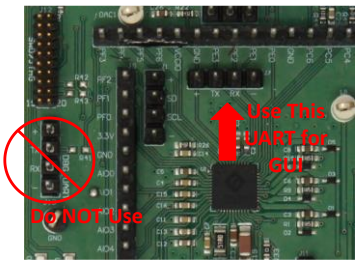
To connect the UART, use a 4 wire cable to connect the PAC EVK 4-pin UART header labelled [+ TX RX -] to the ET-UARTISO-1 4-pin UART header labelled [V T R G]. Note that V T R G stands for Vcc, Tx, Rx, and Gnd respectively. Then, connect the USB port to a PC.

Note:

1) The ET-UARTISO-1 can be plugged directly into a USB port on the PC, or used with a USB extension cable.

The image on the right shows the connection of the ET-UARTISO-1 to the PAC5223 EVK using a USB extension cable.





Note: On PAC55xx EVKs make sure the cable is not connected to the DBG_UART. The image shown on the left highlights the 2 UART headers on the PAC5523 EVK. The DBG_UART is primarily used for printfs() via UART to a PC terminal applications.

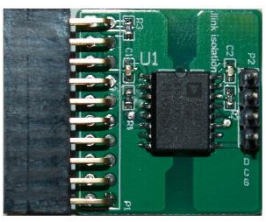
LEDTX and LEDRX LEDs will turn on and off based on UART TX and RX activity respectfully.

When using the ET-UARTISO-1, there are 2 options for SWD connectivity to the PAC EVK:

1) Use the ET-UARTISO-1 SWD adapter board
2) Connect the SWD debugger to the PAC EVK using jumper wires

2.3.2.1 ET-IARISO-1

For SWD debugger connecitivity, the ET-IARISO-1 isolated SWD adapater board can be used to adapt the 4 pin SWD connector of PAC EVKs to most SWD debuggers that have a standard ARM 20-pin IDC connector. Two versions of the ET-IARISO-1 adapter board are shown on the right. The green one has a right angle connector, which can be directly plugged into the SWD debugger without a cable. The blue one has a straight header, which the debugger 20-pin ribbon cable can be plugged into.



Because the debugger side of the ET-IARISO-1 requires power, either

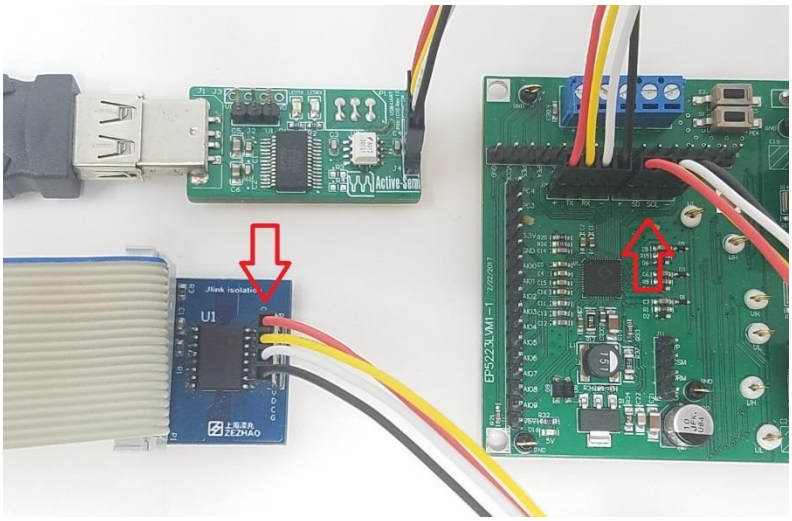
1) The SWD debugger must be cable of supplying 3.3V power on Pin 1 of the 20-pin IDC connector.
2) Or, an external 3.3v supply must be connected to Pin 1 of the 20-pin IDC connector

If these two options aren't possible, then consider the SWD debugger jumper method in paragraph 2.3.2.2.

Note:

1) Debuggers that have built in isolation are not supported, because the 2 isolation circuits will interfere with each other.
2) Note that Pin 1 on the ET-IARISO-1 has a square solder connection on the bottom of the adapater board.

Now, use a 4 wire cable to connect the PAC EVK 4-pin SWD header labelled [+ SD SCL -] to the ET-UARTISO 4-pin UART header labelled [V D C G]. V D C G stands for Vcc, SD, SCL, Gnd respectively. The image below shows the connection for the PAC5223 EVK.



Now connect the 20-pin ribbon cable of the SWD debugger to the ET-IARISO-1 board. Pin 1 of the ET-IARISO board is at the top and represented by the blue wire of the ribbon cable in the figure above. Connect the other end of the 20-pin ribbon cable to the 20 pin IDC connector on the SWD debugger.

Now proceed to paragraph 3 to build and flash the PAC firmware image.

2.3.2.2 SWD Debugger Connection Using Jumper Wires

Use of jumper wires to connect the SWD Debugger should be considered if any of the following are true:

1) an ET-UARTSWD adapter is not available
2) an ET-UARTISO-1 is not available
3) an ET-UARTISO-1 is available, but 3.3V can't be supplied to pin 1 of the adapter
4) The SWD Debugger has built in isolation

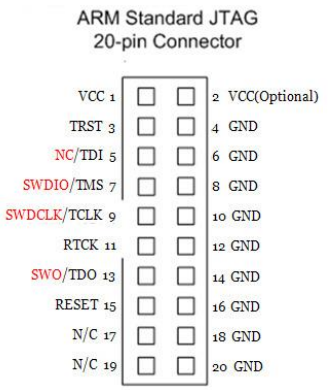
-
-

-

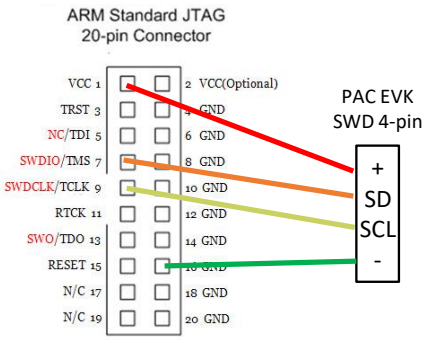
IMPORTANT Notes:

1) When connecting a non isolated SWD Debugger to a high voltage PAC EVK using jumpers, care must be taken or injury and PC damage could occur. For PAC52xx EVKs, the default SWD signal voltage is 5V.
2) If the SWD debugger doesn't support 5V operation, do NOT connect the debugger via this jumper method.

Connection of the SWD Debugger using jumper wires involves connecting 4 jumpers to the PAC EVK. A standard ARM JTAG/SWD 20-pin IDC connector pin out is shown below.



Connect VCC, SWDIO, SWCLK, and a GND using jumper wires to the 4 signals of the PAC EVK 4-pin SWD header labelled [+ SD SCL -]. This is illustrated in the block diagram on the right.



The picture below shows the actual connection of a J-Link debugger and the PAC5223 EVK. The jumper wires in the picture are as follows:

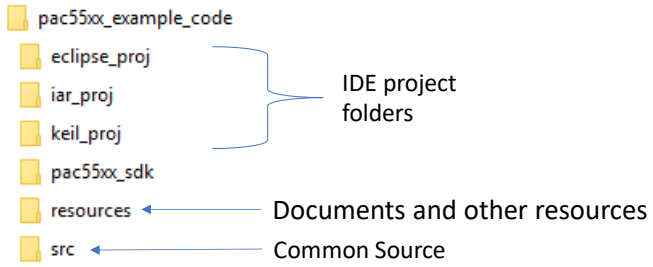
- Pin 1 - Vcc = Red
- Pin 7 - SWDIO = Orange
- Pin 9 - SWCLK = Yellow
- Pin 15 - Gnd = Green

3 Build and Flash the Firmware

The next few steps will configure, build, and flash the firmware to the device on a PAC EVK.

3.1 Choose Device Family and IDE Project

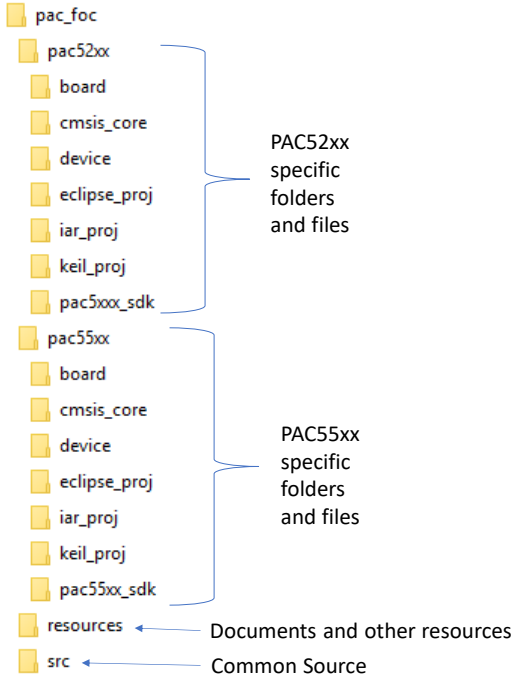
Most PAC firmware contains common source code used for various PAC devices and several IDEs; the common source code is typically stored in the src folder. The figure on the right shows the folder structure of the PAC55xx Example Code.



Sometimes a resources folder is provided that contains documentation and other resources such as a GUI or setting files.

If the firmware only supports a single device family, then the ide project folders will typically be at the top level. Chose the desired IDE and open the project in the associated folder.

If multiple device families are supported by the firmware release, then there will be device family specific code and projects within the device family folders. The figure to the right shows the FOC folder hierarchy for PAC firmware that includes multiple PAC device families. First choose the device family, either PAC52xx or PAC55xx. Then, navigate down to the project folder for one of the IDEs: Keil, IAR, or Eclipse. Using the desired IDE, open the project contained in the associated folder.



3.2 Configure Firmware

The firmware typically needs to be configured for the particular PAC device and/or board. It may also have various settings or features that need to be configured. To configure the firmware, navigate within the folder view of the IDE project to the src folder to reveal main.c and configuration files with “config” or “select” in the file name depending on the firmware release. By editing these files, the appropriate PAC device and/or board can be chosen along with other configuration settings.

3.3 Build and Flash the Firmware

To build and flash the firmware image, follow the steps below:

- From the IDE build the firmware image
  - If building for a PAC55xx device, make sure to install the IDE support files that can be found on active-semi.com in any PAC55xx device product folder on the software tab.
- Verify the build is successful with no errors before moving to the next step.
  - Some IDEs will produce some warnings which are ok to ignore.
- Make sure the Power Supply is turned on and the IDE is configured for the debugger being used
- From the IDE start a debug session to connect to the PAC and flash the firmware image to the device for single stepping, viewing memory/registers, and running the firmware.

Optionally, just flash the firmware without debugging. Note that some IDEs won't reset the PAC device after Flashing and this has to be done manually to start the PAC executing. If this is the case, turn the power supply off and back on to reset the PAC device.

Now the PAC will be running and either performing printfs to the UART, or waiting for communication from a GUI, if the firmware project uses one.

Contact Information

For the latest specifications, additional product information, worldwide sales and distribution locations:

Web: www.qorvo.com

Tel: 1-844-890-8163

Email: customer.support@qorvo.com

Important Notice

The information contained herein is believed to be reliable; however, Qorvo makes no warranties regarding the information contained herein and assumes no responsibility or liability whatsoever for the use of the information contained herein. All information contained herein is subject to change without notice. Customers should obtain and verify the latest relevant information before placing orders for Qorvo products. The information contained herein or any use of such information does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other intellectual property rights, whether with regard to such information itself or anything described by such information. **THIS INFORMATION DOES NOT CONSTITUTE A WARRANTY WITH RESPECT TO THE PRODUCTS DESCRIBED HEREIN, AND QORVO HEREBY DISCLAIMS ANY AND ALL WARRANTIES WITH RESPECT TO SUCH PRODUCTS WHETHER EXPRESS OR IMPLIED BY LAW, COURSE OF DEALING, COURSE OF PERFORMANCE, USAGE OF TRADE OR OTHERWISE, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.**

Without limiting the generality of the foregoing, Qorvo products are not warranted or authorized for use as critical components in medical, life-saving, or life-sustaining applications, or other applications where a failure would reasonably be expected to cause severe personal injury or death.