iND83209 JLINK Interface Guide
V1.0

1 REVISION HISTORY

Table 1 Revision History

Rev #	Date	Action	By
0.1	06/03/2022	First draft	Steve Randlett
1.0	06/06/2022	Initial release, clarify JLINK pinout, make text editable for Segger .xml file modification	Steve Randlett

2 TABLE OF CONTENTS

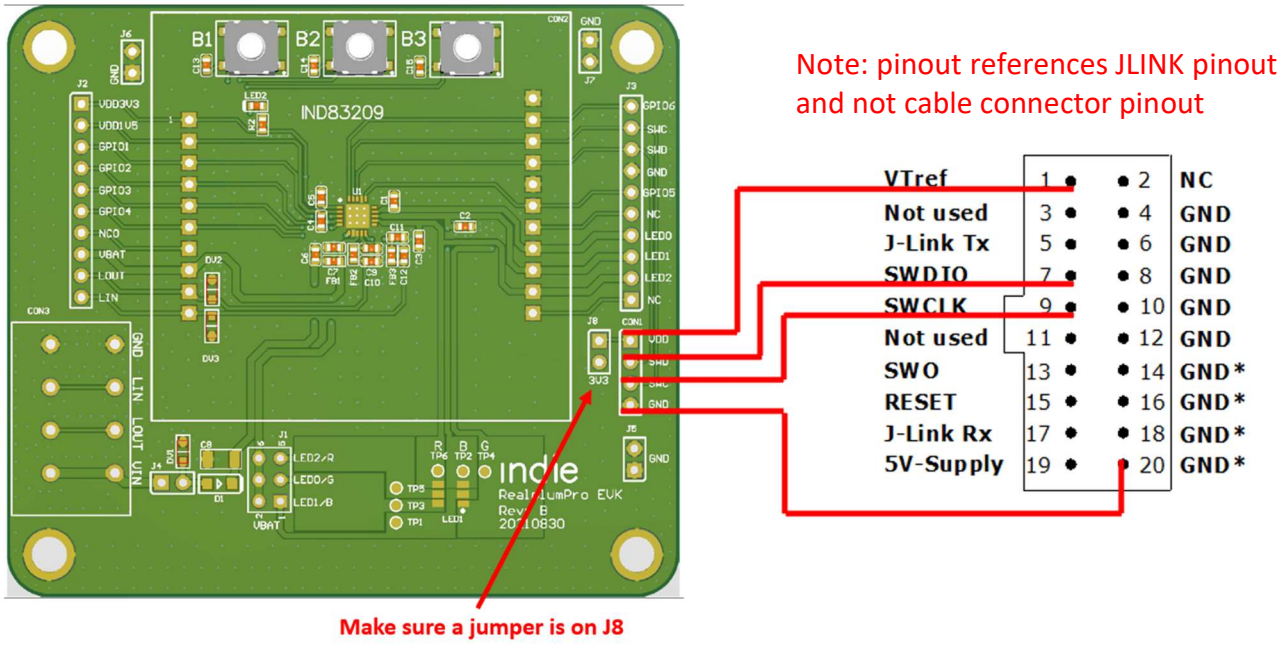
3 BASIC INTRODUCTION

The purpose of this design guide is to aid hardware/software designers in interfacing a Segger JLINK/Flasher debug to the iND83209 EVK or a product PCBA using the iND83209.

To program or debug the iND83209 4 connections to the PCBA are required: SWIO, SWCLK, 3.3V and ground. The Segger tools typically have a standard 20 pin header so an adapter harness is required to connect the EVK. Programming can be accomplished by using the Segger JLINK software. Debugging is accomplished through using the IAR ARM IDE.

3.1 SEGGER JLINK/FLASHER CONNECTION TO EVK

Shown below are the required connections from the Segger JTAG connector:



If a jumper is not placed on J8 then please add one. The VTref pin requires 3.3V since the data and clock pins operate at 3.3V on the iND83209. On a small custom PCBA it is possible to omit the 3.3V connection provided an external 3.3V source is connected to VTref.

3.2 PROGRAMMING/DEBUG THE IND83209

Once the proper connections are made then the iND83209 can be programmed using the Segger tools or debugged through IAR. Please refer to the 'IAR Environment Build User Guide' for further information on configuring the IAR environment.

The Segger tool can be found here:

SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace

After installing the Segger tools you will need to install the iND83209 processor files so it will know how the processor is configured. The iND83209 uses the Verne-M0 core:

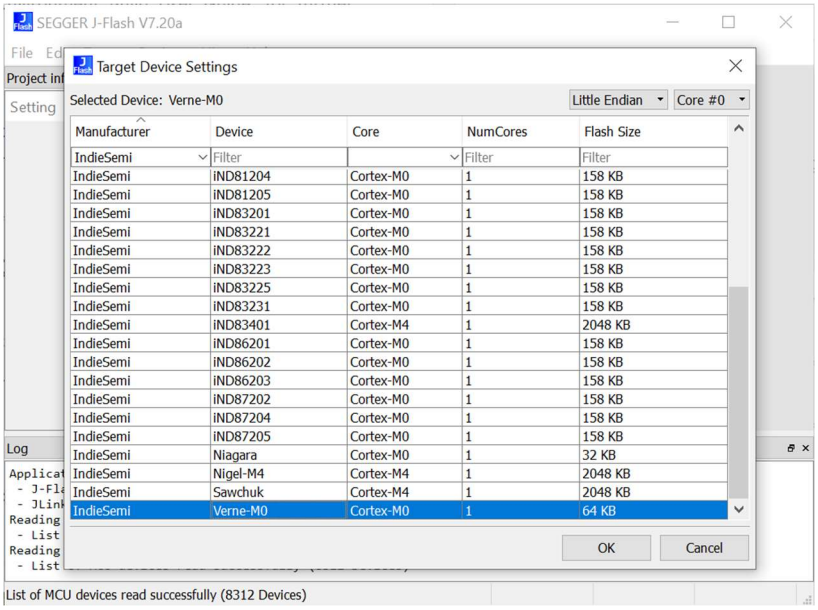
Here are the instructions for adding the Verne-M0 processor files to the Segger file structure on your PC.

1) Find the Segger JLink folder. On my PC it is located at "C:\Program Files (x86)\SEGGER\JLink"
2) Locate the file "JLinkDevices.xml" and open this file for editing in Notepad++ for example. You may need to open it with administrator rights.
3) Add the following lines to the JLinkDevices.xml file and save.

```
  <Device>
    <ChipInfo Vendor="IndieSemi"
       Name="Verne-M0" Core="JLINK_CORE_CORTEX_M0" WorkRAMAddr="0x20000000" WorkRAMSize="0x2000"/>
    <FlashBankInfo
         Name="Internal Flash" BaseAddr="0x00000000" NaxSize="0x0010000"
Loader="Devices/IndieSemi/M0/Indie_Verne_64K.elf"
  </Device>
```

4) Enter the Devices folder
5) Create a new folder and name it "IndieSemi" (if not already existing)
6) In the "IndieSemi" folder, create a folder named M0 (if not already existing)
7) Enter the M0 folder and save the attached "Indie_Verne_64K.elf" file there. Typically the path to this file is "C:\Program Files (x86)\SEGGER\JLink\Devices\IndieSemi\M0\Indie_Verne_64K.elf"

To test the connection using Segger tools launch J-Flash application, create a new project and select the IndieSemi Verne-M0



Power on the EVK and then select ‘Target’ and then ‘Connect’ and you should see something similar to below if your connection is correct. *Please note that the iND83209 has an initial 8 second timeout period (safety precaution to avoid unintended access to debug port) so if you are unable to connect try disconnecting power (allowing enough time for capacitors on battery line to discharge) and connect again within 8 seconds of power up. Once the connection is made the debug port will not timeout.*