



RISC-V SoC-Based Human Presence Detection Using CrossLink-NX VVML Board Demonstration

User Guide

FPGA-UG-02131-1.0

May 2021

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Acronyms in This Document	4
1. Introduction	5
2. Functional Description	5
3. Demo Setup	7
3.1. Hardware Requirements	7
3.2. Software Requirements	7
4. Programming the Demo	8
4.1. Programming the CrossLink-NX Voice and Vision SPI Flash	8
4.1.1. Erasing the CrossLink-NX Voice and Vision Machine Learning Prior to Reprogramming	8
4.1.2. Programming the CrossLink-NX Voice and Vision Machine Learning Board	10
4.1.2.1. Programming the Generated .bit File	10
4.1.2.2. Programming the SensAI Firmware .bin File	12
5. Observing UART Output on Lattice Propel SDK	14
References	16
Technical Support Assistance	17
Revision History	18

Figures

Figure 2.1. Top View of CrossLink-NX Voice and Vision Machine Learning Board	5
Figure 2.2. Bottom View of CrossLink-NX Voice and Vision Machine Learning Board	6
Figure 3.1. Lattice CrossLink-NX Voice and Vision Board	7
Figure 4.1. Radiant Programmer – Default Screen	8
Figure 4.2. Radiant Programmer – Device Selection	8
Figure 4.3. Radiant Programmer – Erase Previous Content	9
Figure 4.4. CrossLink-NX VVML Board – SW5 and SW4 Buttons	10
Figure 4.5. Radiant Programmer – Device Properties to Flash Bit File	11
Figure 4.6. Radiant Programmer – Output Console	12
Figure 4.7. Radiant Programmer – Device Properties to Flash Bin File	13
Figure 5.1. Lattice Propel SDK Default Window	14
Figure 5.2. Opening a Terminal in Propel SDK	15
Figure 5.3. Lattice Propel Launch UART Terminal	15

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
FPGA	Field-Programmable Gate Array
ML	Machine Learning
SD	Secure Digital
SDK	Software Development Kit
SoC	System on Chip
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver-Transmitter
VVML	Voice and Vision Machine Learning

1. Introduction

This document describes the Human Presence Detection demo flashing process for CrossLink™-NX RISC V FPGA SoC platform.

The demo involves mainly two binary components, RTL bitstream and RISC-V firmware, which is written in the SPI flash. The Lattice Radiant™ programmer is used to flash these two binaries on the CrossLink-NX VVML board.

2. Functional Description

The Human Presence Detection Demo is designed to utilize the Lattice Voice and Vision Machine Learning board.

Figure 2.1 and Figure 2.2 show the top view and bottom view of the Voice and Vision board used in this demonstration.

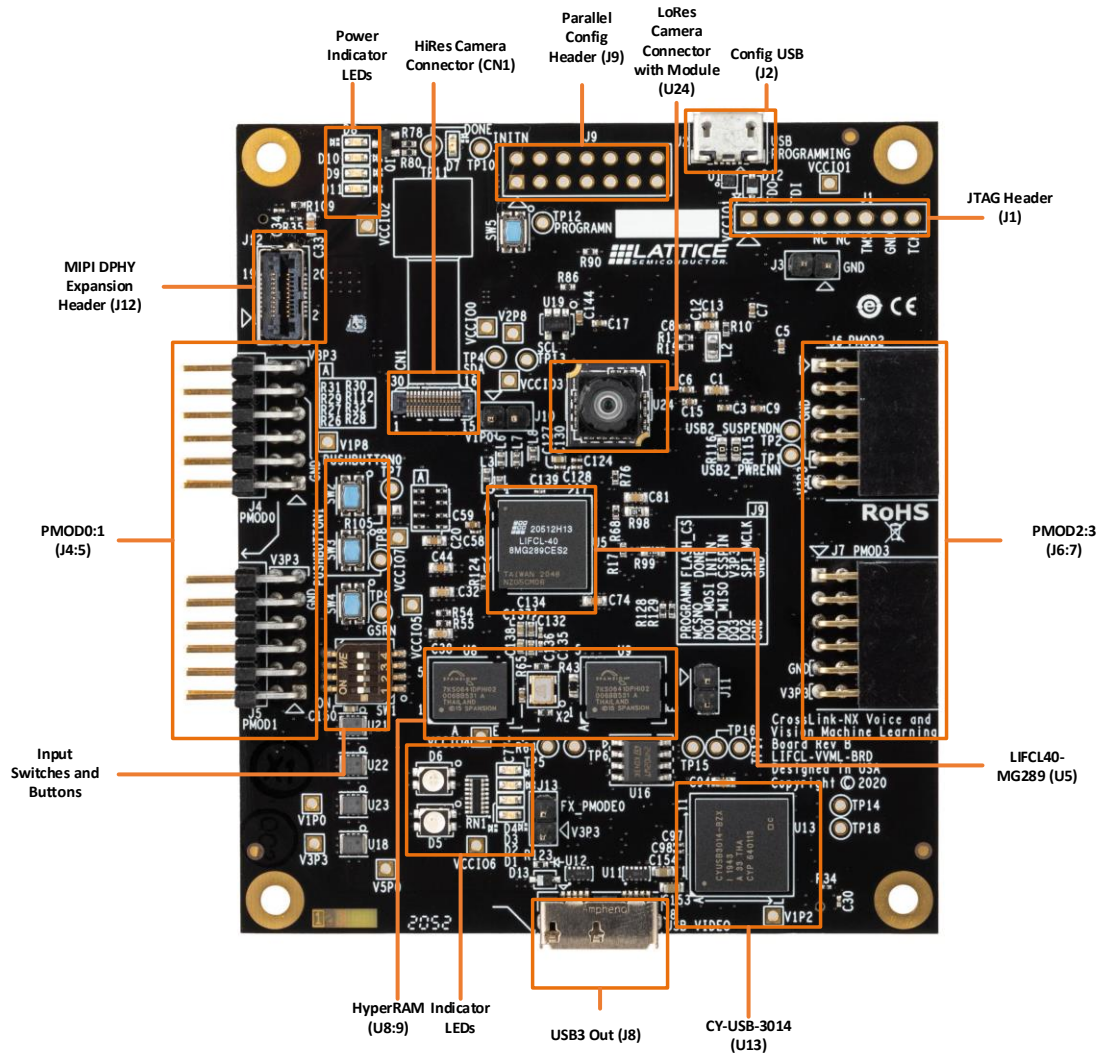


Figure 2.1. Top View of CrossLink-NX Voice and Vision Machine Learning Board

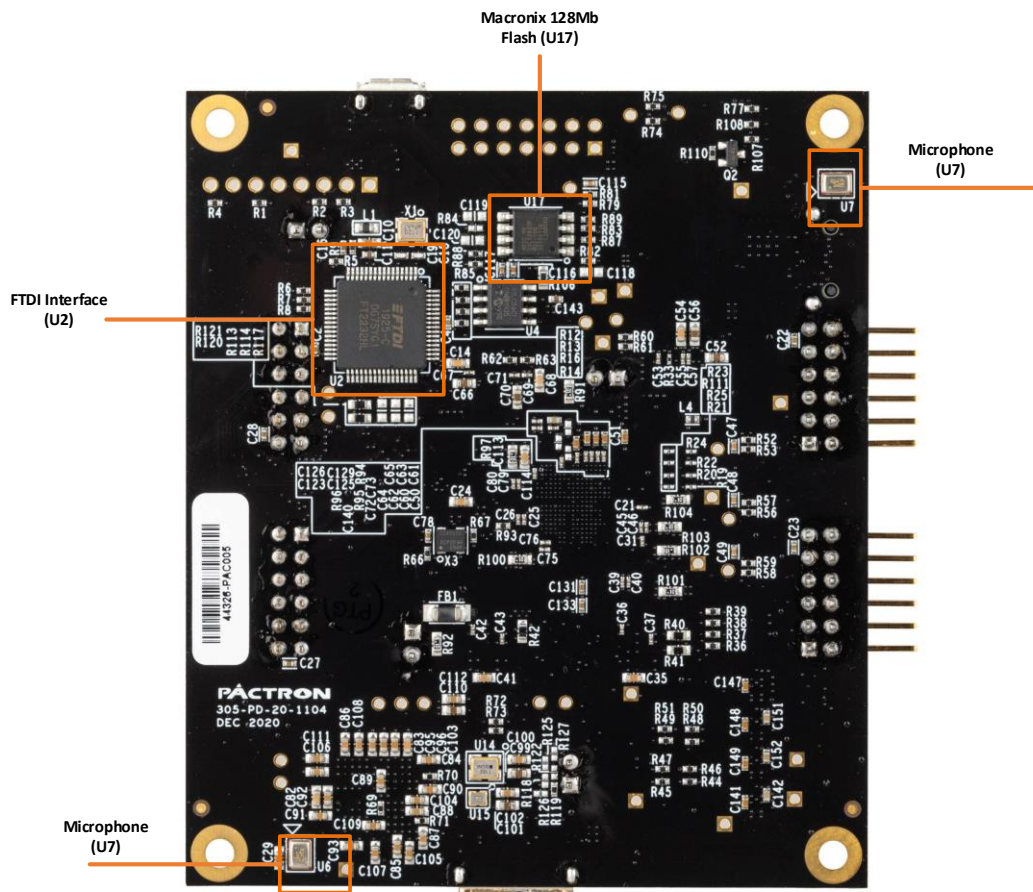


Figure 2.2. Bottom View of CrossLink-NX Voice and Vision Machine Learning Board

3. Demo Setup

This section describes the demo setup.

3.1. Hardware Requirements

- CrossLink-NX Voice and Vision Board

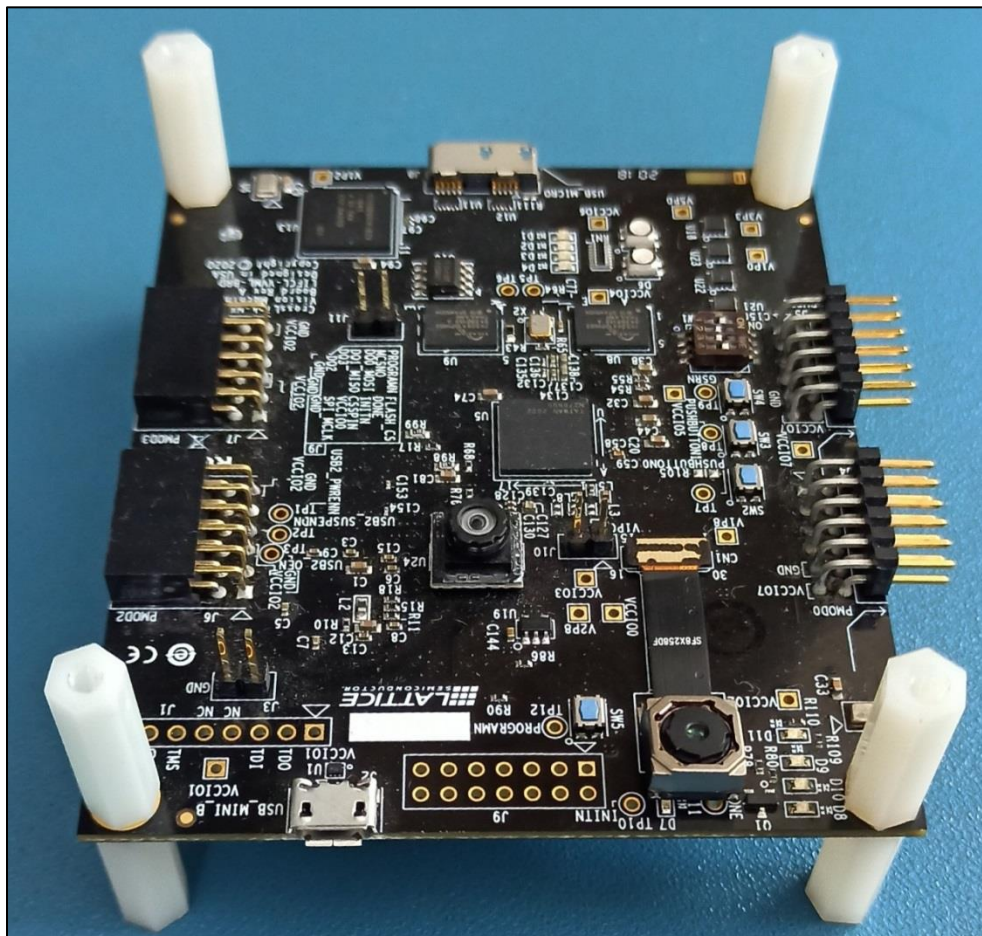


Figure 3.1. Lattice CrossLink-NX Voice and Vision Board

3.2. Software Requirements

- Lattice Radiant Programmer version 2.2 – Refer to <http://www.latticesemi.com/programmer>
- Lattice Propel™ SDK version 2.0 – Refer to <http://www.latticesemi.com/LatticePropel>

4. Programming the Demo

4.1. Programming the CrossLink-NX Voice and Vision SPI Flash

4.1.1. Erasing the CrossLink-NX Voice and Vision Machine Learning Prior to Reprogramming

If the CrossLink-NX Voice and Vision device is already programmed, follow this procedure to first erase the data programmed in the CrossLink-NX Voice and Vision Machine Learning (VVML) board before re-programming the newly generated .bit file with the required .bin file to SPI Flash. If you are doing this, keep the board powered when re-programming the SPI Flash (so it does not reload on reboot).

To erase the CrossLink-NX Voice and Vision Machine Learning board:

1. Start Lattice Radiant Programmer. In the Getting Started dialog box, select **Create a new blank project**.

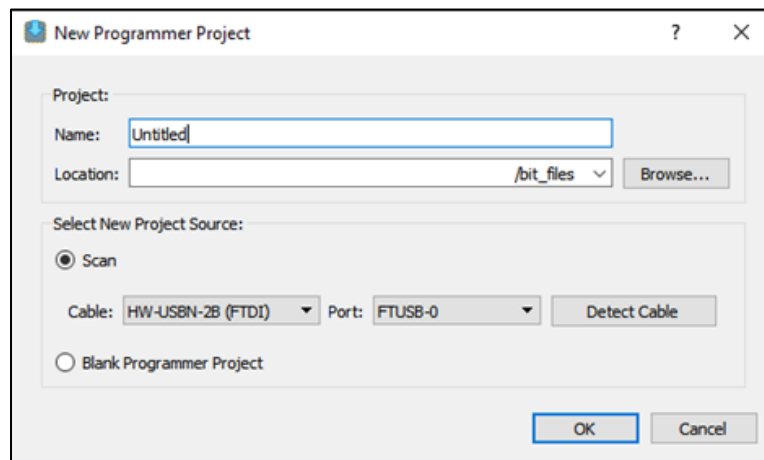


Figure 4.1. Radiant Programmer – Default Screen

2. Click **OK**.
3. Select **LIFCL** for **Device Family** and **LIFCL-40** for **Device** as shown in Figure 4.2.

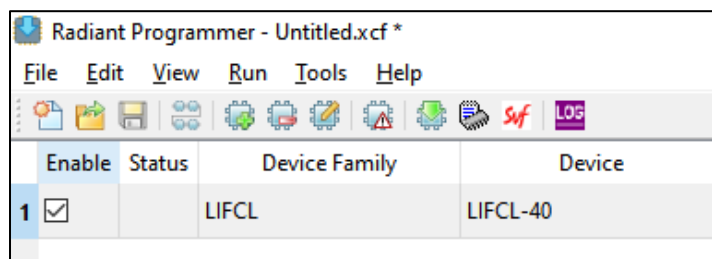


Figure 4.2. Radiant Programmer – Device Selection

4. Right-click on **Operations** and select **Device Properties**.
5. Select the options shown in Figure 4.3 to erase the previously programmed content.

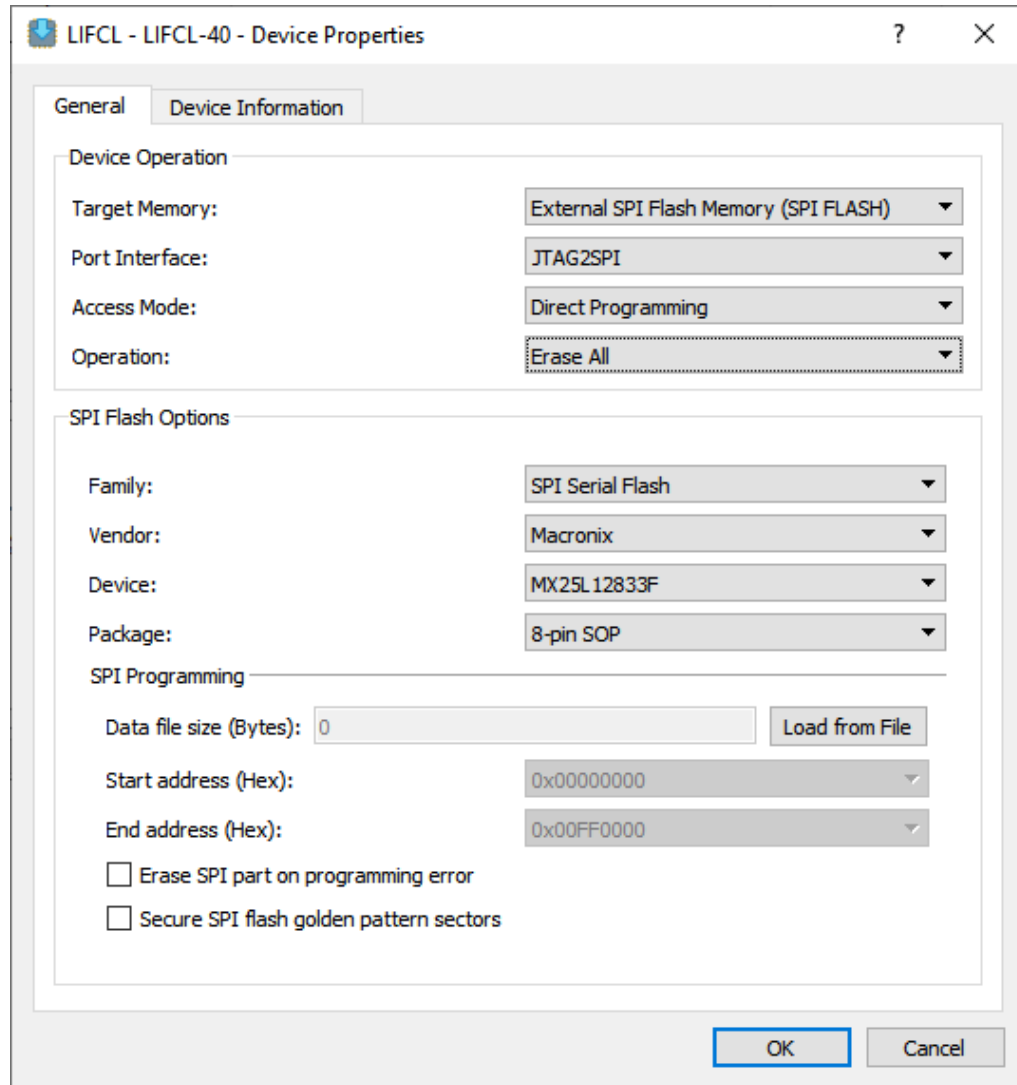


Figure 4.3. Radiant Programmer – Erase Previous Content

6. Click **OK** to close the Device Properties dialog box.
7. Press the **SW5** push button on the board.
8. Click the **Program** button to start the erase operation. Hold it until you see the *Operation Successful* message in the output console.

The **SW5** program button and the **SW4** reset button on the CrossLink-NX VVML board are shown in Error! Reference source not found..

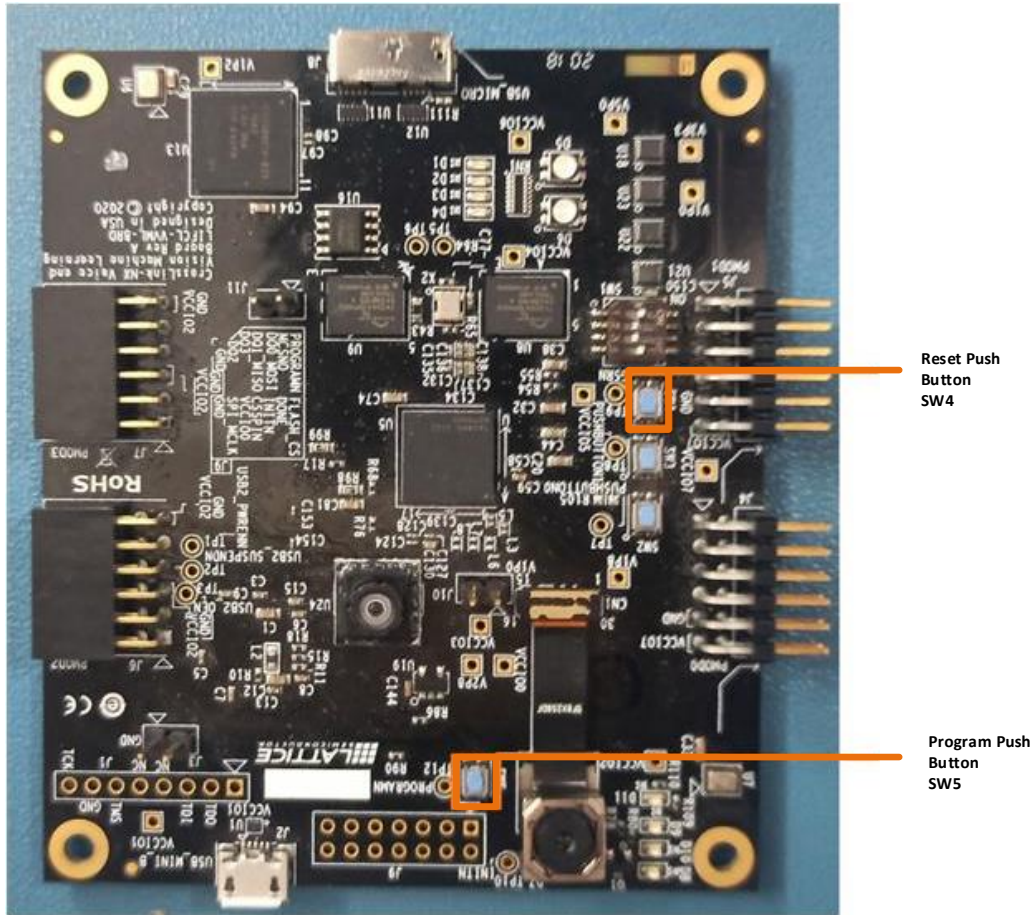


Figure 4.4. CrossLink-NX VVML Board – SW5 and SW4 Buttons

4.1.2. Programming the CrossLink-NX Voice and Vision Machine Learning Board

The following sections provide the steps to program the generated .bit file and the firmware .bin file to the VVML board.

4.1.2.1. Programming the Generated .bit File

To program the generated bitstream file to the CrossLink-NX VVML board:

1. Ensure that the CrossLink-NX VVML board is erased before performing the steps below.
2. In the Radiant Programmer main interface, right-click on **Operation** and select **Device Properties**.
3. Apply the settings below:
 - a. Under Device Operation, select the options below:
 - **Port Interface – JTAG2SPI**
 - **Target Memory – SPI FLASH**
 - **Access Mode – Direct Programming**
 - **Operation – Erase, Program, Verify**
 - b. Under Programming Options, select the generated **soc_main_system_impl_1.bit** file from the **impl1** folder for the **Programming File**.
 - c. For **SPI Flash Options**, make the selections as shown in Figure 4.5.

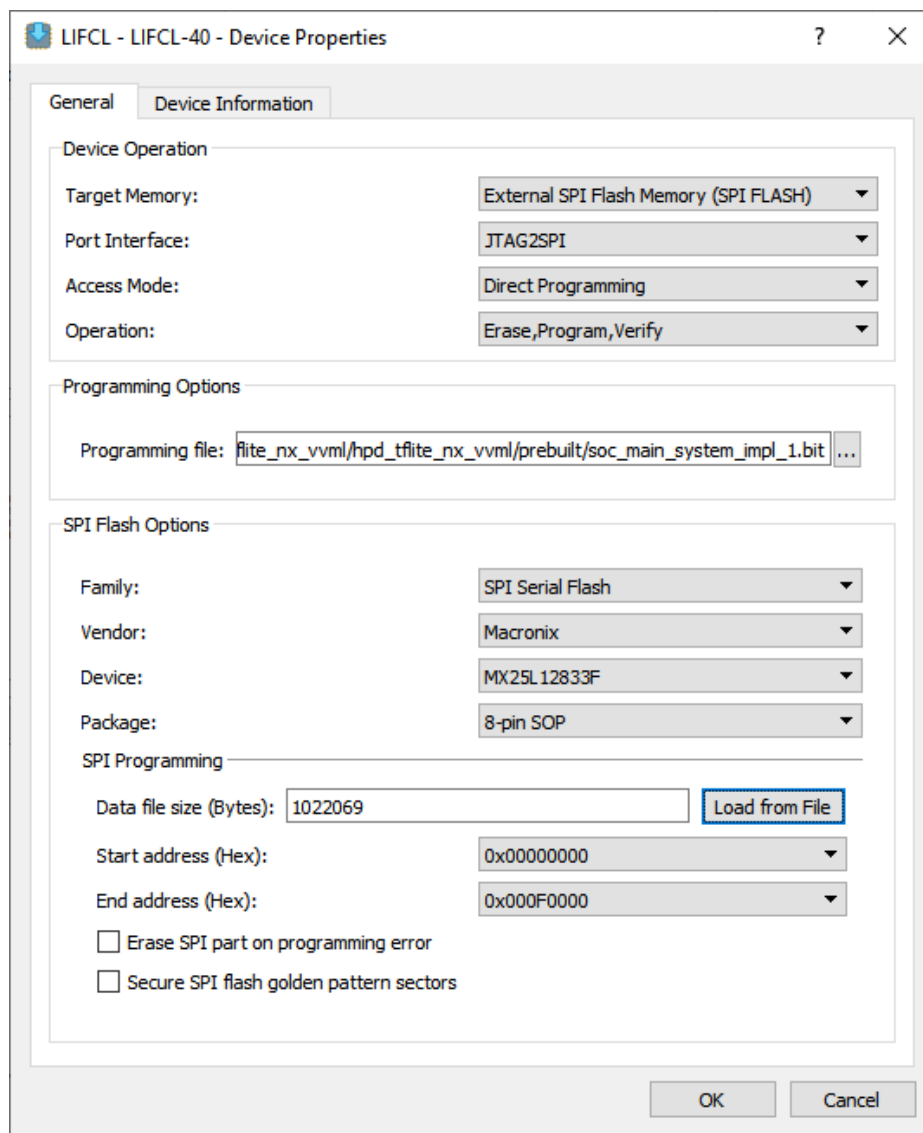


Figure 4.5. Radiant Programmer – Device Properties to Flash Bit File

- d. Click **Load from File** to update the data file size (Bytes) value.
- e. Ensure that the following addresses are correct and Click OK.
 - **Start Address (Hex) – 0x00000000**
 - **End Address (Hex) – 0x000F0000**
4. Click **OK**.
5. Press the **SW5** push button on the board. Click the **Program** button to start the Bit Flash operation, and hold it until you see the *Operation Successful* message in the Radiant log window.
6. After successful programming, the **Output** console displays the result as shown in [Figure 4.6](#).



Figure 4.6. Radiant Programmer – Output Console

4.1.2.2. Programming the SensAI Firmware .bin File

To program the Lattice SensAI™ firmware .bin file to the CrossLink-NX VVML board:

1. In the Radiant Programmer main interface, right-click on **Operation** and select **Device Properties**.
2. Apply the settings below:
 - a. Under Device Operation, select the options below:
 - **Port Interface – JTAG2SPI**
 - **Target Memory – SPI FLASH**
 - **Access Mode – Direct Programming**
 - **Operation – Erase, Program, Verify**
 - b. Under Programming Options, select the **riscv-blink.bin** file from the *prebuilt* folder for the **Programming File**.
 - c. For **SPI Flash Options**, make the selections as shown in [Figure 4.7](#).

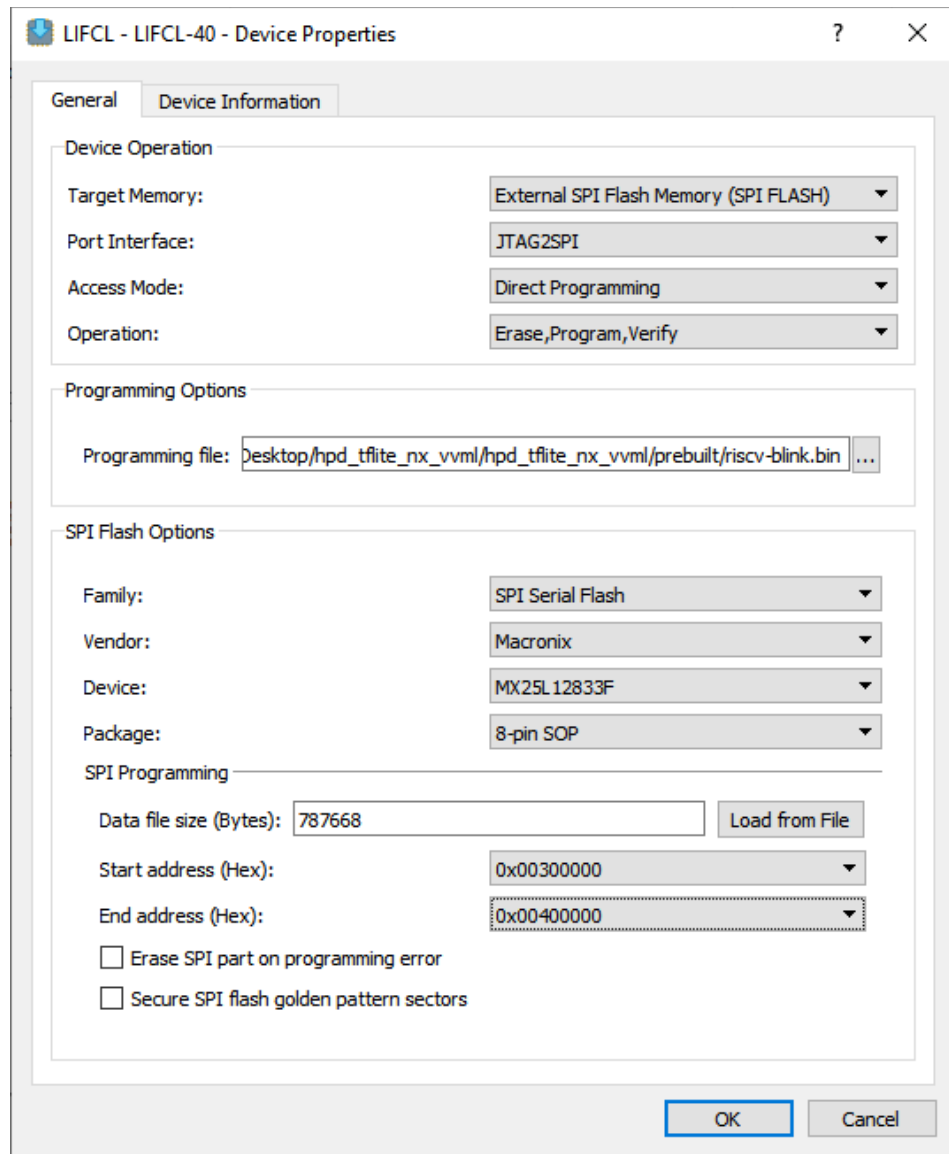


Figure 4.7. Radiant Programmer – Device Properties to Flash Bin File

- d. Do not click on the **Load from File** option.
- e. Ensure that the following addresses are correct:
 - Start Address (Hex) – **0x00300000**
 - End Address (Hex) – **0x00400000**
3. Click **OK**.
4. Press the **SW5** push button on the board. Click the **Program** button to start the Bin Flash operation, and hold it until you see the *Operation Successful* message in the Radiant log window.

Note: After programming the Bit and Bin files, refer to steps provided in the [Observing UART Output on Lattice Propel SDK](#) section to observe the UART output. In case nothing gets printed in the UART terminal, erase the old contents loaded in the board as mentioned in [Erasing the CrossLink-NX Voice and Vision Machine Learning Prior to Reprogramming](#) section, and try following the steps given in [Programming the Generated .bit File](#) by changing to the option to **Erase,Program,Verify Quad 1** in **Device Operation > Operation**. This option enables the SPI on the board and UART outputs are visible. This option is used only once, if UART output is unavailable. You should follow the normal steps provided in [Programming the Generated .bit File](#) onwards to load the .bit file.

5. Observing UART Output on Lattice Propel SDK

This section provides the steps to observe the human presence output in the UART terminal using the Lattice Propel SDK software.

To observe the UART output:

1. After the .bit file and .bin files are programmed to the VVML CrossLink-NX board, power OFF the board.
2. Open the Lattice Propel SDK version 2.0 software tool.
3. The Lattice Propel launch window opens and prompts for the **Workspace** location. Launch the tool.
4. Select the **Terminal** option available as shown in [Figure 5.1](#).
5. If the Terminal window is not visible after opening the Lattice Propel SDK tool, you can open the terminal from the toolbar **Window > Show view > other > Terminal (folder) > Terminal**.

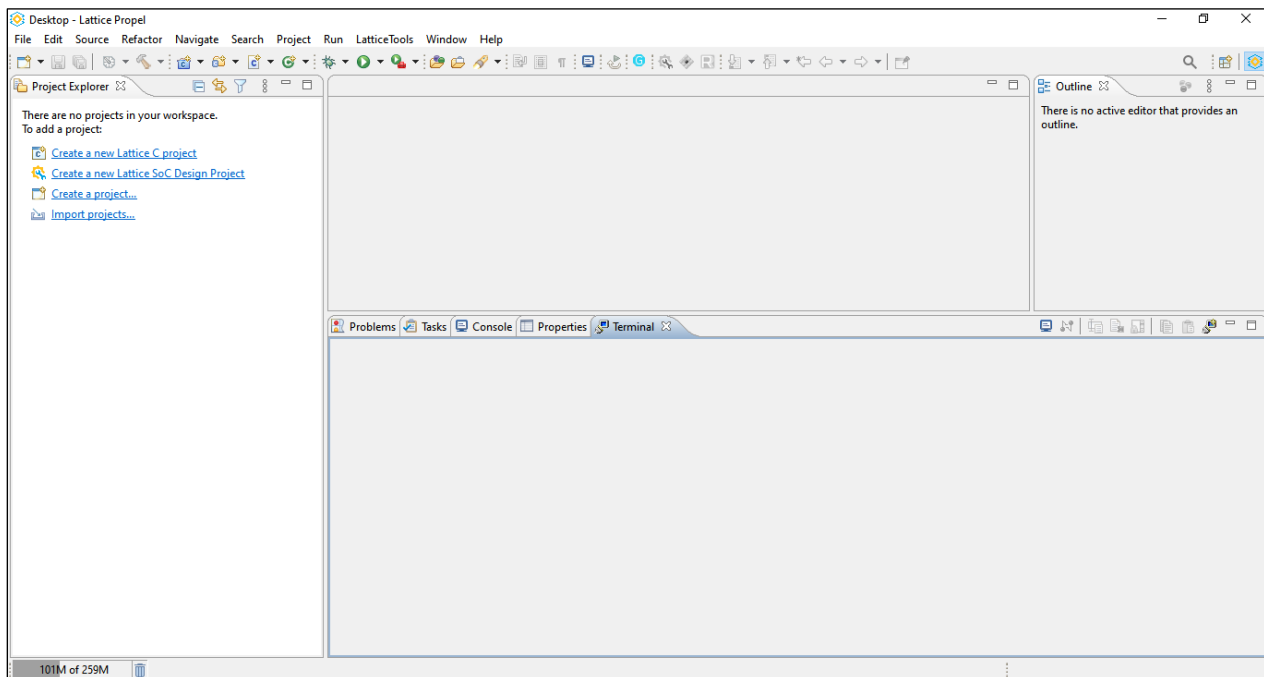


Figure 5.1. Lattice Propel SDK Default Window

6. Click the **Launch Terminal** button to open a terminal as shown in [Figure 5.2](#). The launch terminal dialog box is displayed.

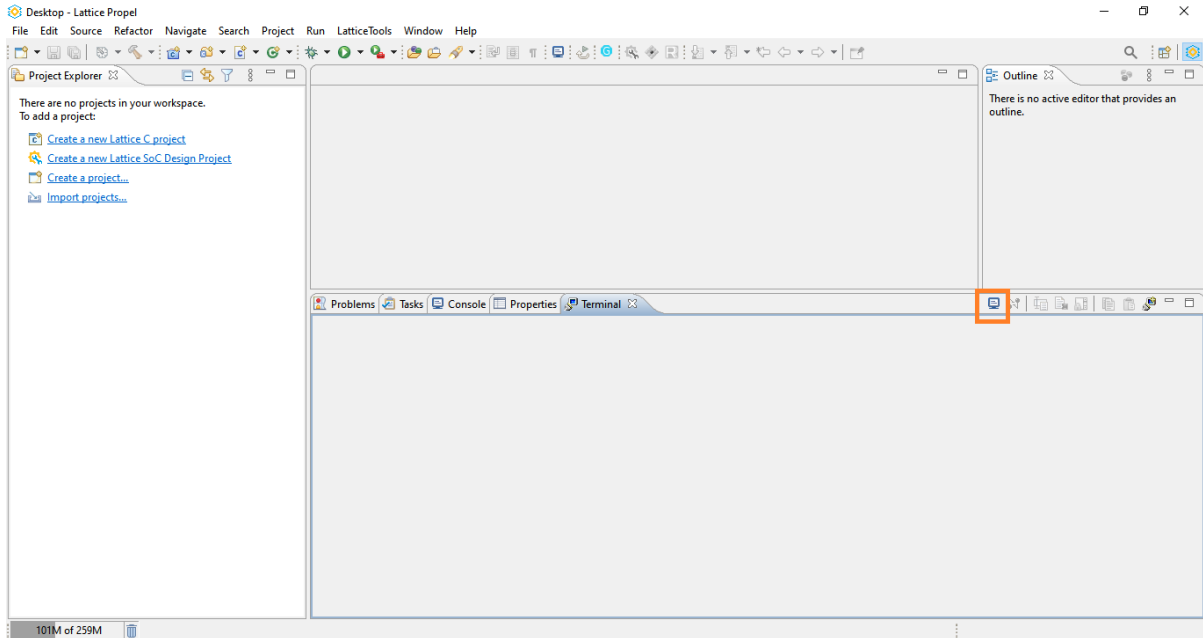


Figure 5.2. Opening a Terminal in Propel SDK

7. Power **ON** the board by connecting to the system.
8. Select the settings in the terminal dialog box as shown in [Figure 5.3](#). Always select the highest numbered COM port available in the **Serial Port** option. For example if COM1, COM6, and COM7 are visible, select **COM7**.

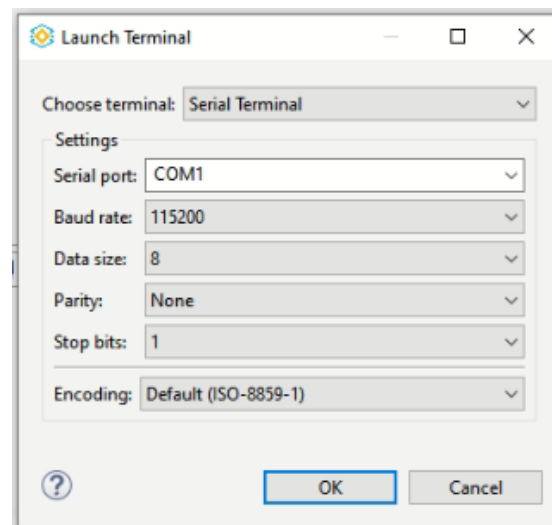


Figure 5.3. Lattice Propel Launch UART Terminal

9. Click **OK**.
10. Reset the board using the **SW4** button and observe the UART values are getting printed in the COM7 space.

References

For more information, refer to the following documents:

- [Lattice Propel Builder 2.0 User Guide](#)
- [Lattice Propel SDK 2.0 User Guide](#)

For more information on the CrossLink-NX Voice and Vision Machine Learning FPGA board, visit

[CrossLink-NX Voice and Vision Machine Learning Board](#) web page.

For complete information on the Lattice Radiant project-based environment, design flow, implementation flow, tasks, and simulation flow, refer to the [Lattice Radiant 2.2 User Guide](#).

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.0, May 2021

Section	Change Summary
All	Initial release



www.latticesemi.com