Human-Machine Interface Demonstration

User Guide

FPGA-UG-02229-2.0

May 2025

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, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. 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. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. 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.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Contents 3
Abbreviations in This Document 4
1. Introduction 5
2. Installation 5
2.1. Installation Requirements 5
2.1.1. Hardware Requirements 5
2.1.2. Software Requirements 5
2.1.3. Binaries 5
2.2. Host Computer Setup 5
2.3. CrossLink-NX-33 VVML Board Setup 6
2.3.1. Connectors Setup 6
2.3.2. Programming the SPI Flash Memory 7
3. Running the Pipeline 11
4. The Pipeline Output 12
4.1. Video 12
4.2. Output Data 12
4.2.1. Data Format 12
4.3. Visualizing the Pipeline Output 14
Reference 15
Technical Support Assistance 16
Revision History 17

Figures

Figure 2.1. Crosslink-NX-33 VVML Board 6
Figure 2.2. Lattice Radiant Programmer New Project Window 7
Figure 2.3. Lattice Radiant Programmer Main Window 7
Figure 2.4. Lattice Radiant Programmer Device Properties Windows 8
Figure 2.5. Lattice Radiant Programmer Device Properties Window – CrossLink-NX-33 VVML Board Configuration 9
Figure 2.6. Lattice Radiant Programmer Device Properties Window – CrossLink-NX-33 VVML Configuration Done 10

Tables

Table 4.1. Pipeline Output Format 12
Table 4.2. Output Data Field Format 12
Table 4.3. Output Person Data Field Format 13
Table 4.4. Output Person Face Format 13
Table 4.5. Output Ideal Person Format 14

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
EVE	Edge Vision Engine
HMI	Human-Machine Interface
PMOD	Peripheral Module Interface
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VVML	Voice and Vision Machine Learning

1. Introduction

This document provides instructions to setup and use the human-machine interface (HMI) demonstration on the Lattice™ CrossLink™-NX-33 Voice and Vision Machine Learning (VVML) board. This HMI pipeline is an embedded solution that detects persons visible in the board camera field of view. The information reported by the pipeline includes position and pose of the detected persons' bodies, faces, distance to the camera and identification status.

2. Installation

2.1. Installation Requirements

2.1.1. Hardware Requirements

- A host computer with at least two universal serial bus (USB) ports, where one USB port must support USB3
- Lattice CrossLink-NX-33 VVML board
- USB3 A to USB3 Micro-B cable
- USB2 A to USB2 Micro-B cable
- The USB to Transistor-Transistor Logic (TTL) adapter that can operate at 3.3 V levels
- Long pins break-away headers, double sided

2.1.2. Software Requirements

You can download the Lattice Radiant™ Programmer for Windows® and Linux from the Lattice Radiant Software web page.

2.1.3. Binaries

The CrossLink-NX-33 VVML board SPI flash memory image binary is located in the Human-Machine Interface Demonstration folder: *FPGA_AI_Firmware_HMI_*.bin*.

2.2. Host Computer Setup

Install the Lattice Radiant Programmer on the host computer. Ensure that your USB to TTL adapter is recognized as a COM port by connecting the adapter to the host computer. The adapter does not need to be connected to the CrossLink-NX-33 VVML board for your operating system to detect it. When you have confirmed the detection, unplug the adapter.

2.3. CrossLink-NX-33 VVML Board Setup

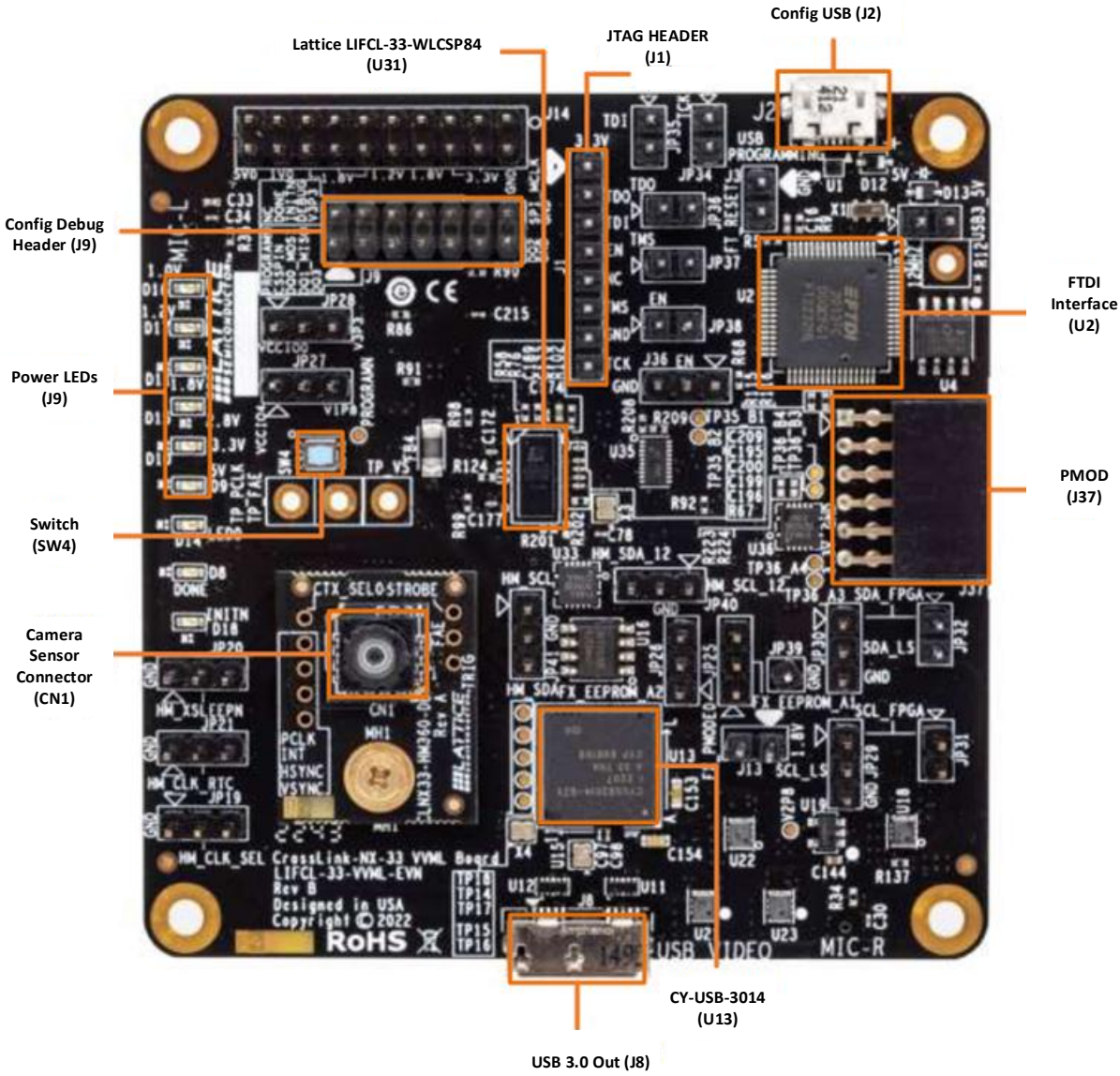

Figure 2.1. Crosslink-NX-33 VVML Board

2.3.1. Connectors Setup

To setup the connectors, follow these steps:

1. Insert the pin headers into the peripheral module interface (PMOD) sockets.
2. Adjust the voltage level of the USB to TTL adapter to 3.3 V.
3. Connect the USB to TTL adapter to the CrossLink-NX-33 VVML board using the PMOD expanded universal asynchronous receiver/transmitter (UART) configuration. Refer to the standard 12-pin male connector placement on Pmod boards in the *Digilent® Pmod™ Interface Specification* from the Diligent web page.
4. Connect the VCC pin of the USB to TTL adapter to PMOD pin 6.
5. Connect the GND pin of the USB to TTL adapter to PMOD pin 5.
6. Connect the RXD pin of the USB to TTL adapter to PMOD pin 4.
7. Connect the TXD pin of the USB to TTL adapter to PMOD pin 3.

8. Connect the USB3 A to USB3 Micro-B cable to the J8 USB 3.0 out socket.
9. Connect the USB2 A to USB2 Micro-B cable to the J2 config USB socket.

2.3.2. Programming the SPI Flash Memory

To program the SPI flash memory, follow these steps:

1. Connect the CrossLink-NX-33 VVML board to the host computer using the USB2 A to USB2 Micro-B cable.
2. Launch the Lattice Radiant Programmer. The **New Project** window as shown in Figure 2.2 is displayed. If the **Cable** option is not set, the CrossLink-NX-33 VVML board is not connected properly to the host computer.

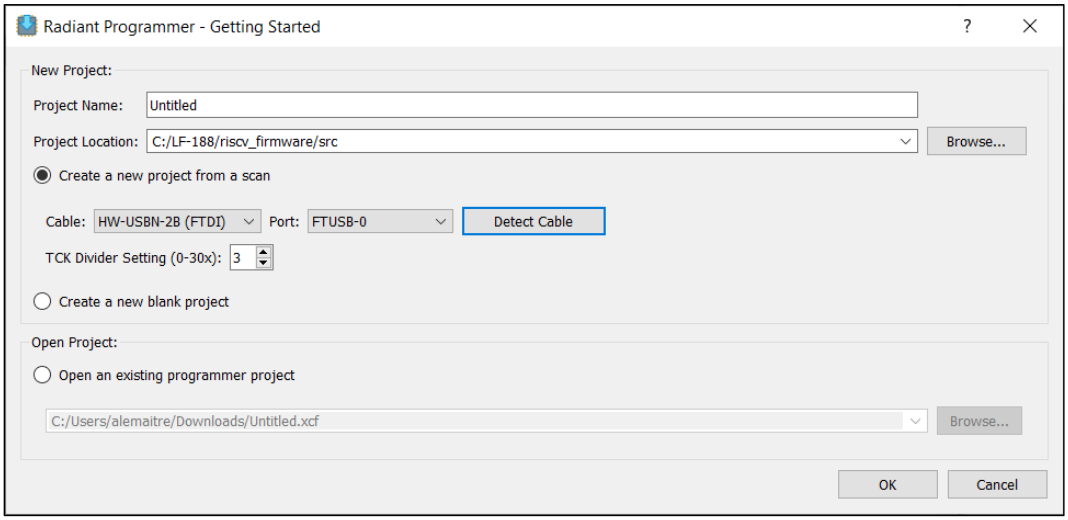

Figure 2.2. Lattice Radiant Programmer New Project Window

3. Select **Create a new project from a scan**, click **OK**. Figure 2.3 is displayed.

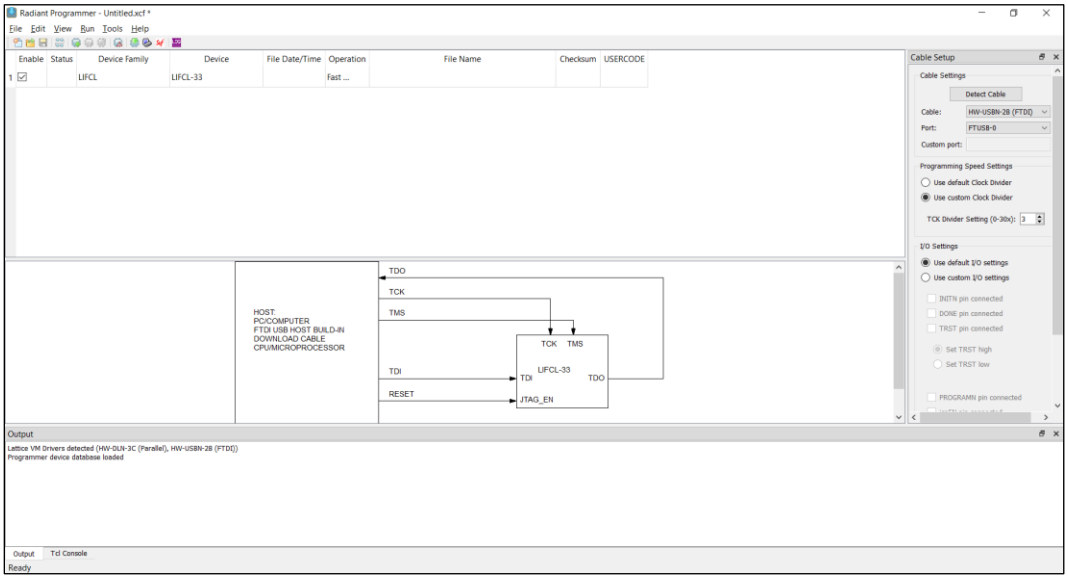

Figure 2.3. Lattice Radiant Programmer Main Window

4. Left click on line 1 to select the device.
5. In the **Edit** menu, select **Device Properties**.
6. The **Device Properties** windows as shown in Figure 2.4 is displayed.

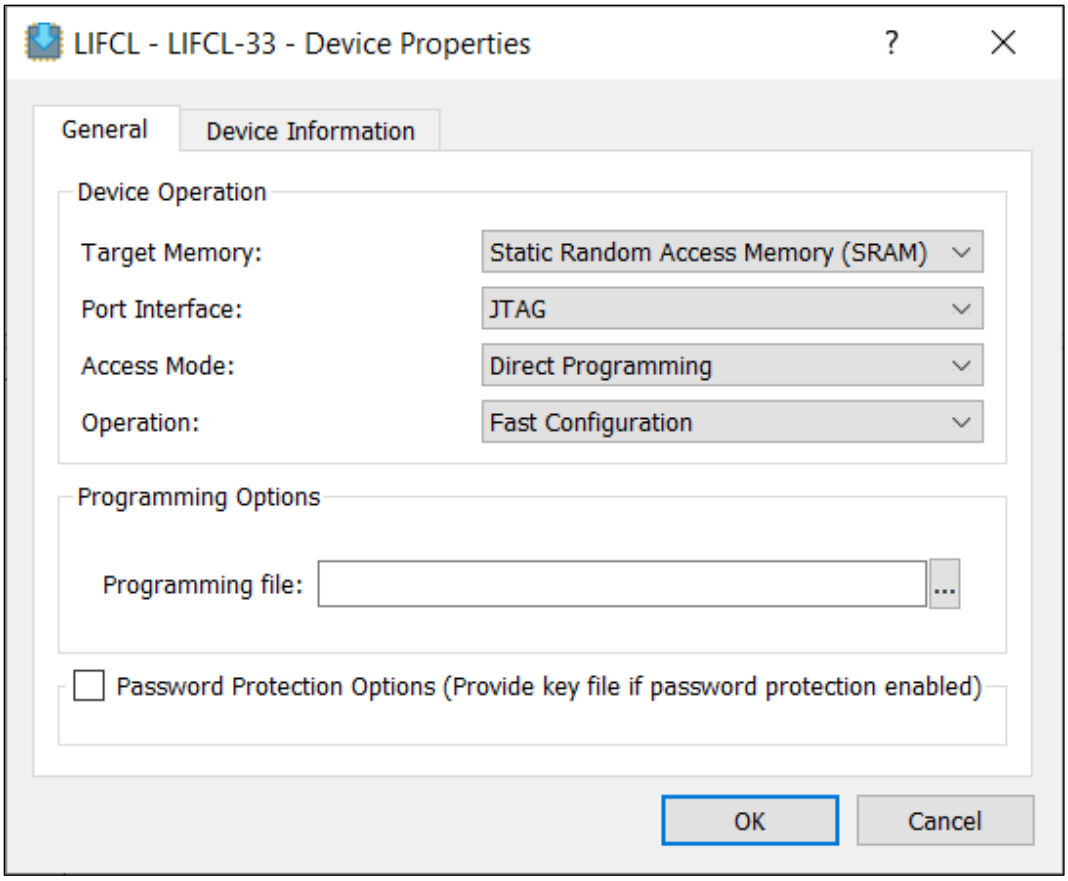

Figure 2.4. Lattice Radiant Programmer Device Properties Windows

7. Set **Target Memory** to **External SPI Flash Memory (SPI FLASH)**. Set all other options as shown in Figure 2.5.

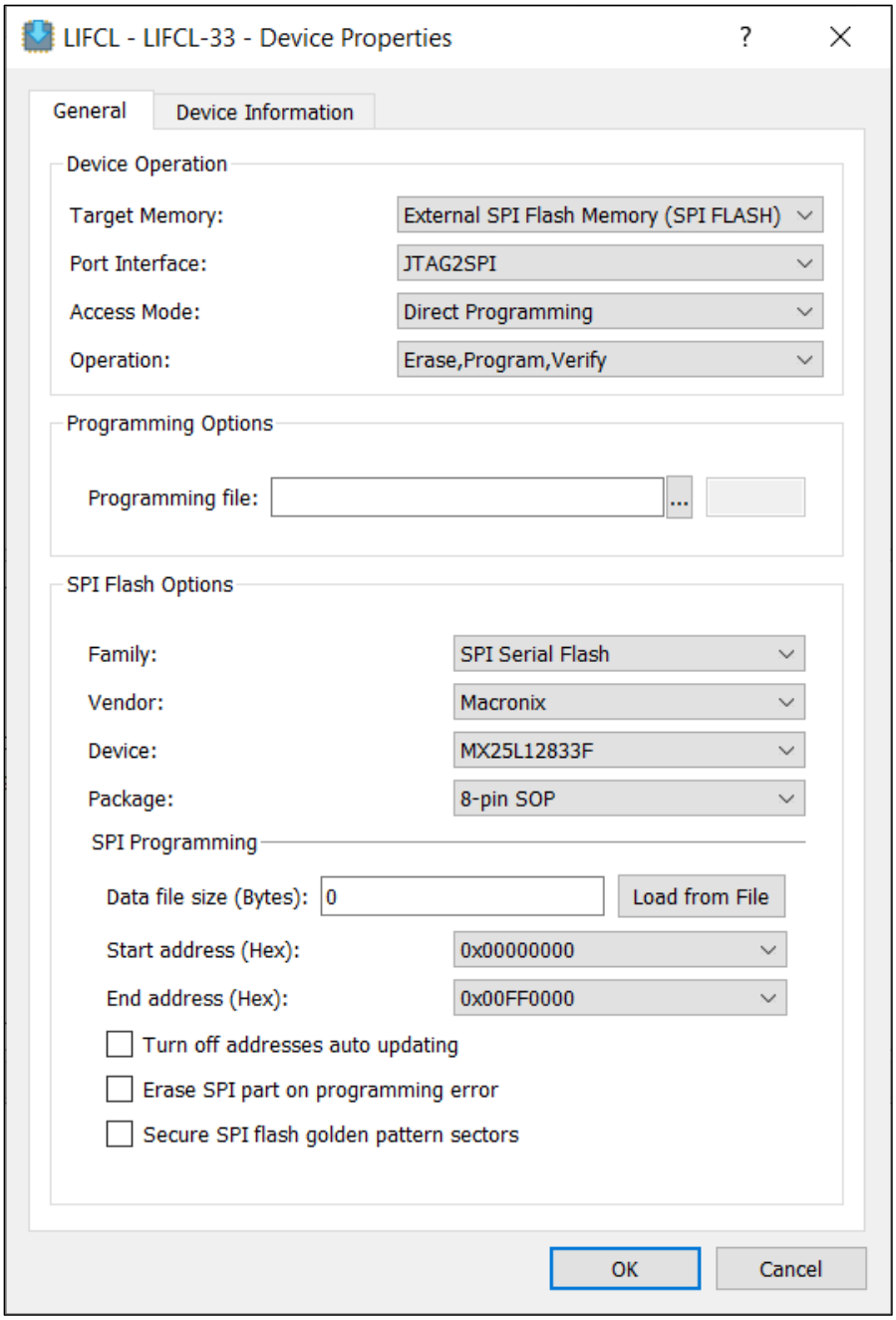

Figure 2.5. Lattice Radiant Programmer Device Properties Window – CrossLink-NX-33 VVML Board Configuration

8. Set **Programming file** to *FPGA_AI_Firmware_HMI_*.bin* as shown in Figure 2.6.

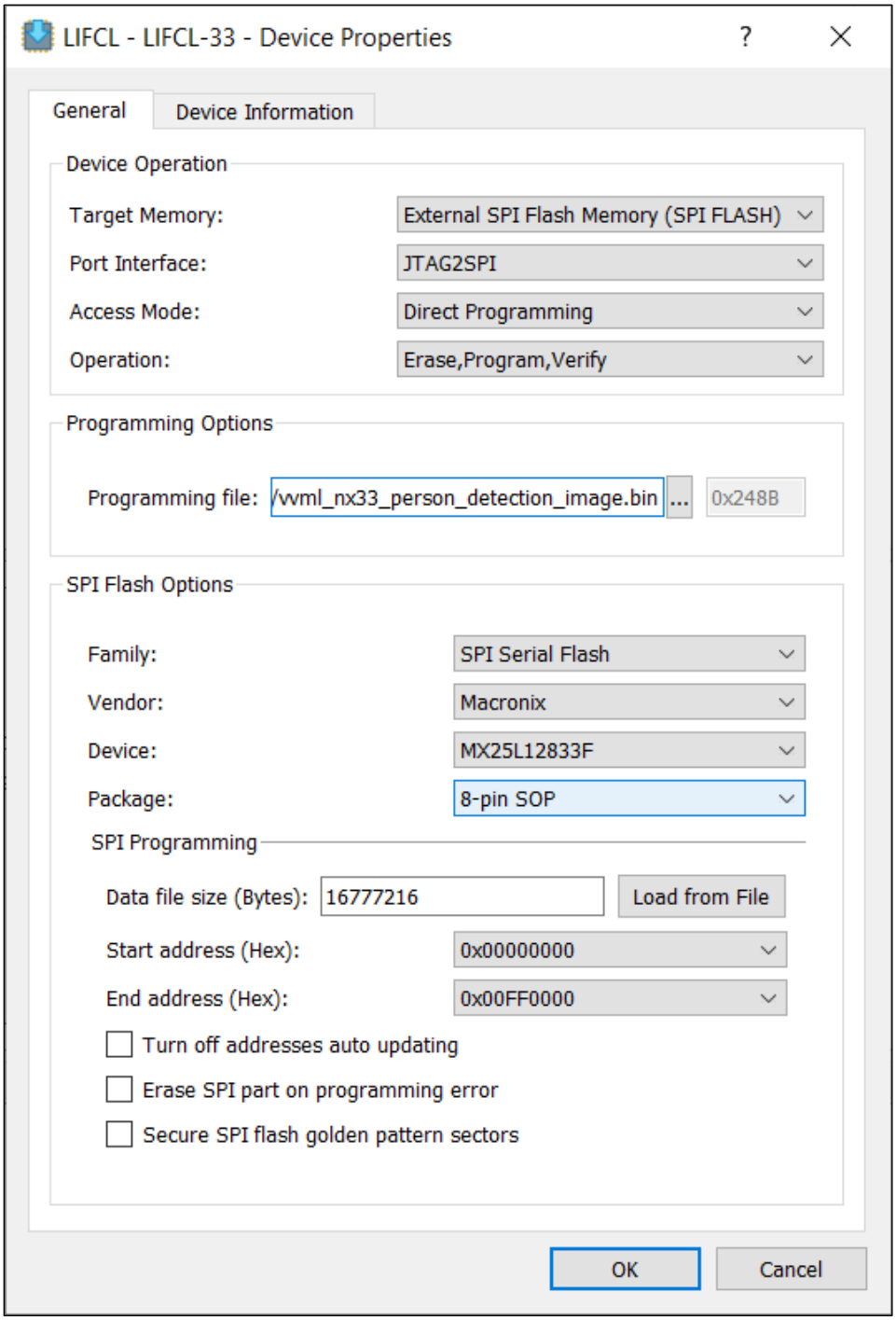

Figure 2.6. Lattice Radiant Programmer Device Properties Window – CrossLink-NX-33 VVML Configuration Done

9. Click **OK**.
10. In the **Run** menu, select **Program Device**. Note that programming can take several minutes.
11. The output window displays a successful completion message. Otherwise, check the connection between the CrossLink-NX-33 VVML board and the host computer and try again.
12. Disconnect the board from the host computer.

3. Running the Pipeline

When your board is properly set up, the pipeline runs automatically when the CrossLink-NX-33 VVML board is powered up. The board can be powered up by plugging the board to the host computer through the USB3 A to USB3 Micro-B cable, the USB2 A to USB2 Micro-B cable, or the USB to TTL adapter, assuming the connector is properly connected to the board. When the pipeline is running, the LEDs on the board are all light up, with LED D10 blue and all others green.

4. The Pipeline Output

4.1. Video

The pipeline transmits the camera video feed via USB3. To capture the video signal on the host computer, connect the CrossLink-NX-33 VVML board using a USB3 A to USB3 Micro-B cable. The board is recognized as an FX3 camera. You can use any compatible application, such as the Windows Camera App or VLC, to capture the video feed.

4.2. Output Data

The pipeline transmits information about detected individuals via UART. To capture this data, connect the CrossLink-NX-33 VVML board to the host computer using the USB to TTL adapter, ensuring the adapter is properly connected to the board as described in the Connectors Setup section.

4.2.1. Data Format

The pipeline binary output must be interpreted properly. The following tables describe the format used.

4.2.1.1. Data Packet Structure

Table 4.1 describes the overall data packet structure.

Table 4.1. Pipeline Output Format

Name	Offset from Previous Field (Byte)	Size (Byte)	Value	Comment
Start flag	0	1	0x7e	—
Data length	1	2	0–65536	Length of the data part of the packet only.
Data	2	0–65536	—	Refer to the Data Field Format section.
Fletcher's checksum	0–65536	2	0–65536	The checksum is computer on the Data part only.

All packets start with a flag. The data length refers only to the data field, excluding the start flag, the data length itself, and the checksum. We utilize Fletcher's checksum to ensure data integrity, which is calculated solely on the data, without including the start flag and data length. The host computer captures the data field, computes the Fletcher's checksum, and compares the data field with the checksum sent by the pipeline. If the two checksums do not match, the data must be discarded.

4.2.1.2. Data Field Format

Table 4.2. Output Data Field Format

Name	Offset from Previous Entry	Size (Byte)	Value	Unit	Comment
Number of persons	0	4	0–4294967296	—	Interpret as unsigned int32
Person 0 data	4	Variable	Variable	—	Refer to the Person Data Field Format section
Person 1 data	4	Variable	Variable	—	Refer to the Person Data Field Format section
Person 2 data	4	Variable	Variable	—	Refer to the Person Data Field Format section
Person 3 data	4	Variable	Variable	—	Refer to the Person Data Field Format section
Person 4 data	4	Variable	Variable	—	Refer to the Person Data Field Format section
Ideal user data	4	Variable	Variable	—	Refer to the Ideal Person Data section

Note: The actual number of person data fields depends on the value of the number of persons' field. If this number is 0, there is no person data field.

4.2.1.3. Person Data Field Format

Table 4.3. Output Person Data Field Format

Name	Offset from Previous Entry	Size (Byte)	Value	Unit	Comment
Person body bounding box confidence	0	4	0–100, with two digits precision	—	Interpret as signed int32, then divide by 1024
Person body bounding box left	4	4	–2147483648 to 2147483647	Pixels from the left edge of the image	Interpret as signed int32
Person body bounding box top	4	4	–2147483648 to 2147483647	Pixels from the top edge of the image	Interpret as signed int32
Person body bounding box right	4	4	–2147483648 to 2147483647	Pixel from the left edge of the image	Interpret as signed int32
Person body bounding box bottom	4	4	–2147483648 to 2147483647	Pixel from the top edge of the image	Interpret as signed int32
Person body pose	4	4	FRONT, SIDE, BACK	—	Interpret as unsigned int32. Value 0 means FRONT, 1 means SIDE, and 2 means BACK
Person distance from camera	4	4	0–4294967296	cm	Interpret as unsigned int32

4.2.1.4. Person Face Data Field

Table 4.4. Output Person Face Format

Name	Offset from Previous Entry	Size (Byte)	Value	Unit	Comment
Person's face bounding box confidence	0	4	0–100	—	Interpret as signed int32 then divide by 1024
Person's face bounding box left	4	4	–2147483648 to 2147483647	Pixels from the left edge of the image	Interpret as signed int32
Person's face bounding box top	4	4	–2147483648 to 2147483647	Pixels from the top edge of the image	Interpret as signed int32
Person's face bounding box right	4	4	–2147483648 to 2147483647	Pixels from the left edge of the image	Interpret as signed int32
Person's face bounding box bottom	4	4	–2147483648 to 2147483647	Pixels from the top edge of the image	Interpret as signed int32
Person's face yaw angle	4	4	–180 to 180	Degrees	Interpret as signed int32 then divide by 1024
Person face pitch angle	4	4	–180 to 180	Degrees	Interpret as signed int32 then divide by 1024

The fields related to a person's face are accessible only when the Person face availability field is set to True.

4.2.1.5. Ideal Person Data

Table 4.5. Output Ideal Person Format

Name	Offset from Previous Entry	Size (Byte)	Value	Unit	Comment
Ideal person index	0	4	0-4	—	Interpret as signed int32. Indicates which user to is the ideal one. This must not be equal or greater than the number of detected persons.
Ideal person identification status	4	4	REGISTERED, UNREGISTERED, UNKNOWN	—	Interpret as signed int32. Value 0 means REGISTERED, 1 means UNREGISTERED, 2 means UNKNOWN.

Note: Ideal person's data is only available if at least one person is detected.

4.3. Visualizing the Pipeline Output

The Lattice sensAI Edge Vision Engine (EVE) tool can be used to visualize the pipeline output overlayed over the camera image output from the USB3 socket of the CrossLink-NX-33 VVML board. The CrossLink-NX-33 VVML board must be connected to the host computer with the USB3 A to USB3 Micro-B cable and the USB to TTL adapter. Follow the instructions in the Lattice sensAI Edge Vision Engine web page to use this feature.

Reference

- CrossLink-NX-33 Voice and Vision Machine Learning Board web page
- Lattice sensAI Edge Vision Engine web page
- Diligent web page, for the Digilent Pmod Interface Specification
- Fletcher's checksum web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 2.0, May 2025

Section	Change Summary
All	• Renamed document from *FPGA AI Firmware Pipeline* to *Human-Machine Interface Demonstration*. • Performed minor formatting and editorial edits.
Abbreviations in This Document	Updated abbreviations.
Introduction	Updated section.
Installation	• Renamed section from *Requirements* to Installation Requirements. • Renamed section from *Hardware* to Hardware Requirements and updated section. • Renamed section from *Software* to Software Requirements and updated section. • Updated the folder for the SPI flash memory image binary in the Binaries section. • Updated the Host Computer Setup section. • Updated the steps in the Connectors Setup section. • Updated the folder of the programming file in the Programming the SPI Flash Memory section.
Running the Pipeline	Updated section.
The Pipeline Output	• Updated the value and comment for *Data* in Table 4.1. Pipeline Output Format. • Updated the unit and comment in Table 4.2. Output Data Field Format. • Updated the unit for *Person body bounding box confidence* and *Person body pose* in Table 4.3. Output Person Data Field Format. • Update the unit for *Person's face bounding box confidence* in Table 4.4. Output Person Face Format. • Updated the unit in Table 4.5. Output Ideal Person Format. • Updated the Visualizing the Pipeline Output section.
References	Updated references.

Revision 1.0, November 2024

Section	Change Summary
All	Initial release.

www.latticesemi.com