CrossLink-NX QVGA Mobilenet Human Identification on VVML Board Demonstration

User Guide

FPGA-UG-02141-1.0

November 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, with all faults and associated risk the responsibility entirely of the 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

Figures

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
EEPROM	Electronically Erasable Programmable Read-Only Memory
FPGA	Field-Programmable Gate Array
GPU	Graphics Processing Unit
I^2C	Inter-Integrated Circuit
QVGA	Quarter Video Graphics Array
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
USB	Universal Serial Bus
VVML	Voice and Vision Machine Learning

1. Introduction

The CrossLink™-NX QVGA Mobilenet Human Identification on VVML Board Demonstration uses machine learning to identify different human faces. A CNN acceleration engine is trained to deliver accurate identification by extracting 256 16-bit characteristics from each registered face.

This demo is targeted to the CrossLink-NX Voice and Vision Machine Learning (VVML) board.

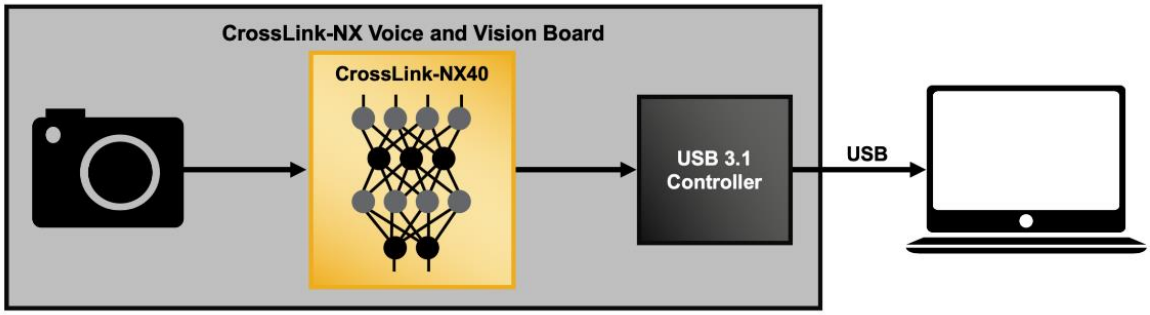

Figure 1.1. Block Diagram

2. Functional Description

This demo registers and identifies faces without the need for retraining. You do not have to upload images and perform lengthy retraining processes using a GPU (Graphics Processing Unit). The demo supports performance of up to 30 frames per second and power consumption of 200 mW on Lattice CrossLink-NX 40k FPGAs.

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

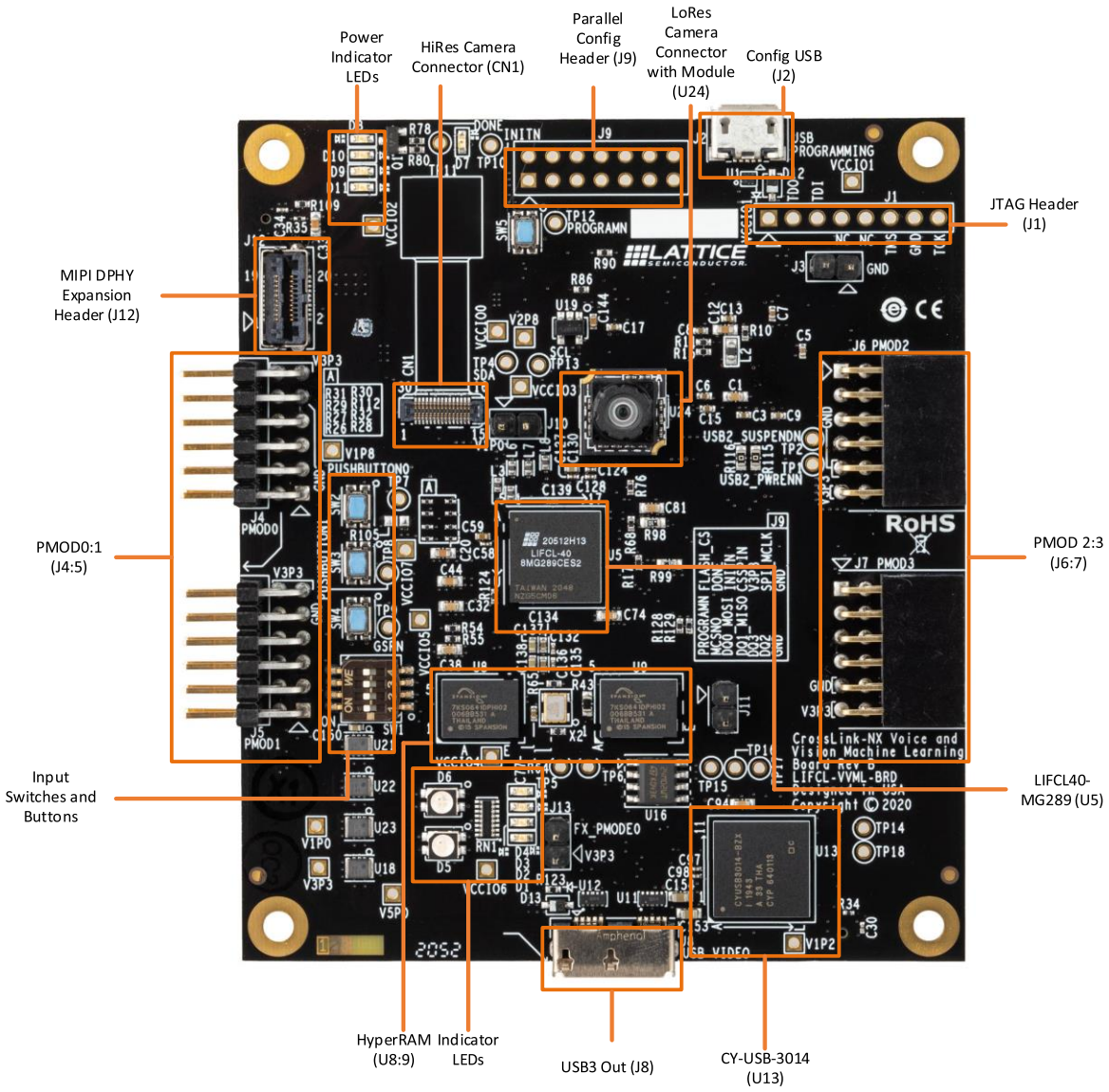

Figure 2.1. Top View of the CrossLink-NX Voice and Vision Machine Learning Board

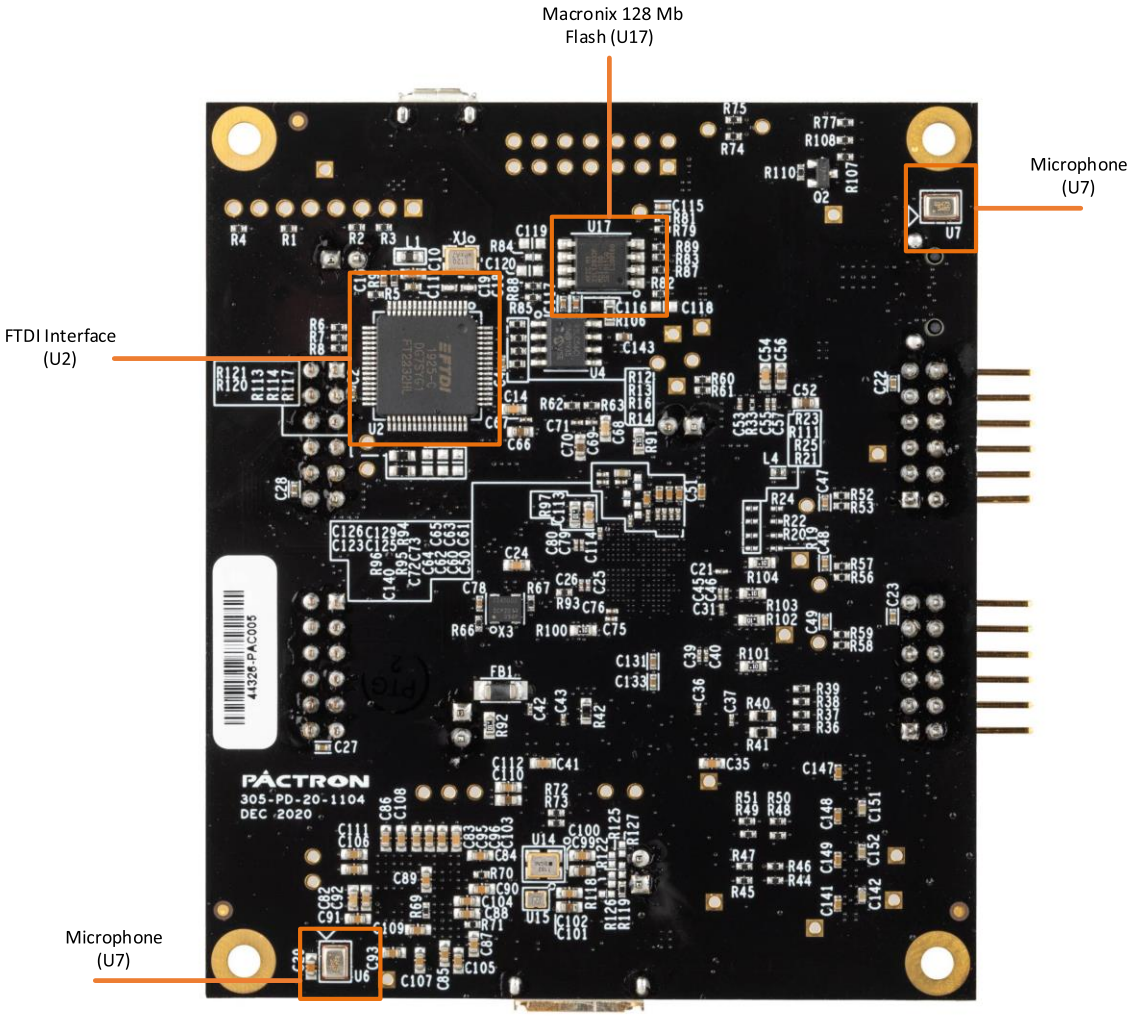

Figure 2.2. Bottom View of the CrossLink-NX Voice and Vision Machine Learning Board

3. Demo Setup

This section describes the demo setup.

3.1. Hardware Requirements

- CrossLink-NX Voice and Vision Machine Learning Board, Rev. B
- USB3 cable
- Personal computer

3.2. Software Requirements

- Lattice Radiant™ Programmer version 3 (Refer to http://www.latticesemi.com/programmer)

4. Programming the Demo

4.1. Package Folder Structure

Figure 4.1 shows the demo folders and files after unzipping the package.

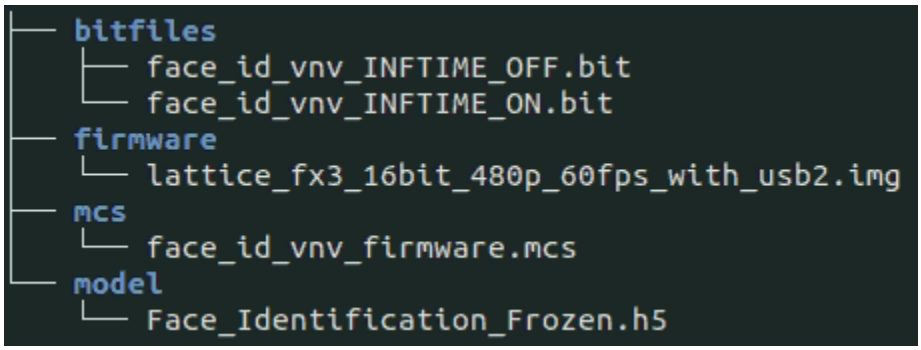

Figure 4.1. Demo Package Folder Structure.

4.2. Loading FX3 Firmware from I²C EEPROM

To load the firmware:

1. Connect the USB3 port of the CrossLink-NX VVML Board (Rev B) to the PC using the USB3 cable. Then connect the Jumper J13 on the board.
2. Open the USB Control Centre application. The Cypress FX3 SDK should be installed.
3. Press the SW2 button to reset the FX3 chip. Figure 4.2 shows the boot loader device screen.

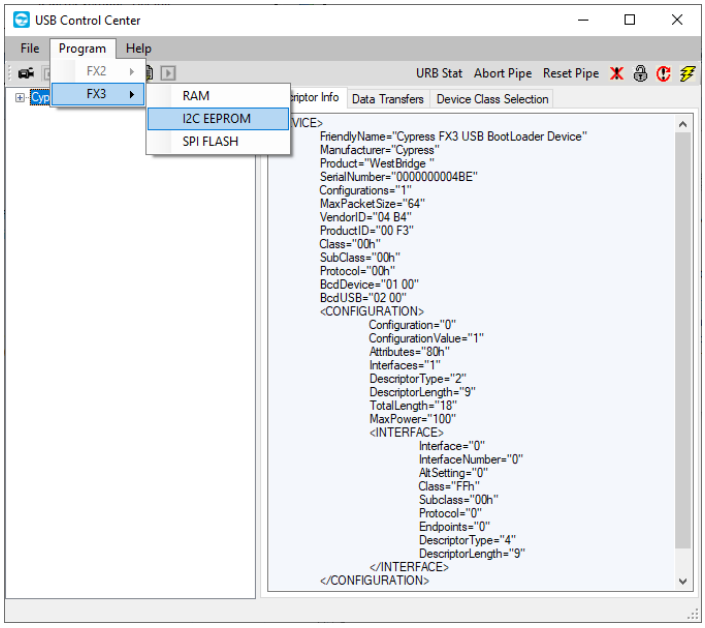

Figure 4.2. Selecting FX3 I²C EEPROM in USB Control Centre

4. Select **Cypress USB Bootloader**.
5. Click **Program > FX3 > I2C E2PROM**.
6. Locate and select the FX3 image file for the 640 × 480p 60 Hz 16 bit configuration.

7. The Firmware is programmed in the I^2C E2PROM.
8. After the operation is completed, a message acknowledging successful programming is shown at the bottom taskbar.
9. Remove jumper J13.
10. Power OFF and then power ON the board.
11. The FX3 boots from I^2C E2PROM.

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

4.3.1. Erasing the CrossLink-NX Voice and Vision SRAM Prior to Reprogramming

If the CrossLink-NX device is already programmed (either directly, or loaded from SPI Flash), follow this procedure first to erase the CrossLink-NX SRAM memory before re-programming the SPI Flash. If you are doing this, keep the board powered when reprogramming the SPI Flash (so it does not reload on reboot).

To erase the CrossLink-NX device:

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

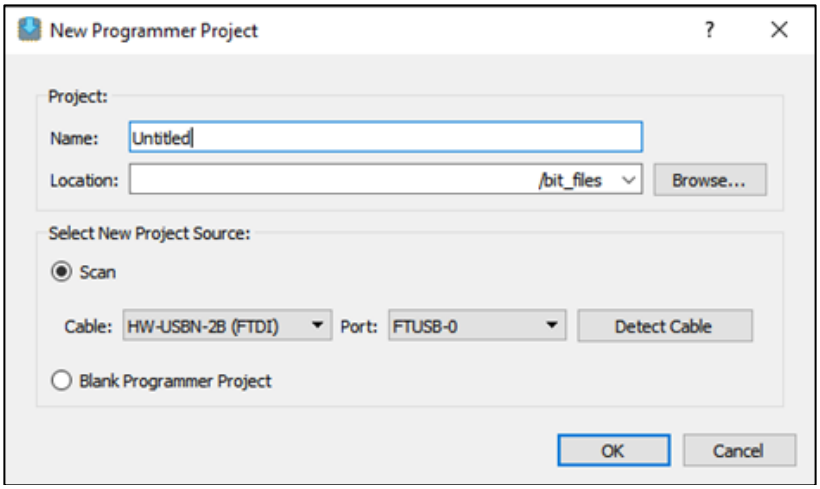

Figure 4.3. Lattice Radiant Programmer Default Screen

2. Click **OK**.
3. In the Lattice Radiant Programmer main interface, select **LIFCL** for Device Family, as shown in Figure 4.4. Select **LIFCL** for Device Vendor, and **LIFCL-40** for Device,

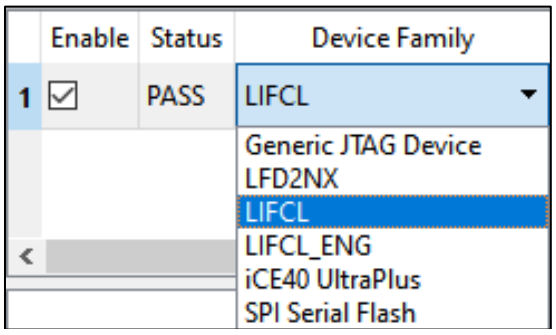

Figure 4.4. Lattice Radiant Programmer- Device Selection

4. Right-click and select **Device Properties**.
5. Select **JTAG** for **Port Interface**, **Direct Programming** for **Access Mode**, and **Erase Only** for **Operation**, as shown in Figure 4.5.

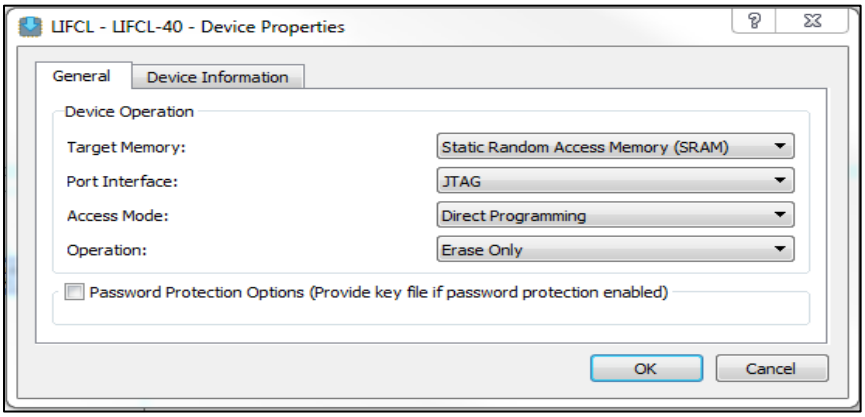

Figure 4.5. Lattice Radiant Programmer – Device Operation

6. Click **OK** to close the **Device Properties** dialog box.
7. Press and hold SW5 until you see the *Successful* message in the Lattice Radiant log window.
8. In the Radiant Programmer main interface, click the Program button to start the erase operation.

4.3.2. Programming the CrossLink-NX VVML Board

To program the CrossLink-NX Voice and Vision SPI Flash:

1. Ensure that the CrossLink-NX Voice and Vision device SRAM is erased by performing the steps in the Erasing the CrossLink-NX Voice and Vision SRAM Prior to Reprogramming section.
2. In the Lattice Radiant Programmer main interface, right-click **Operation** and select **Device Properties**.
3. In the **Device Properties** dialog box, apply the settings shown in Figure 4.6.

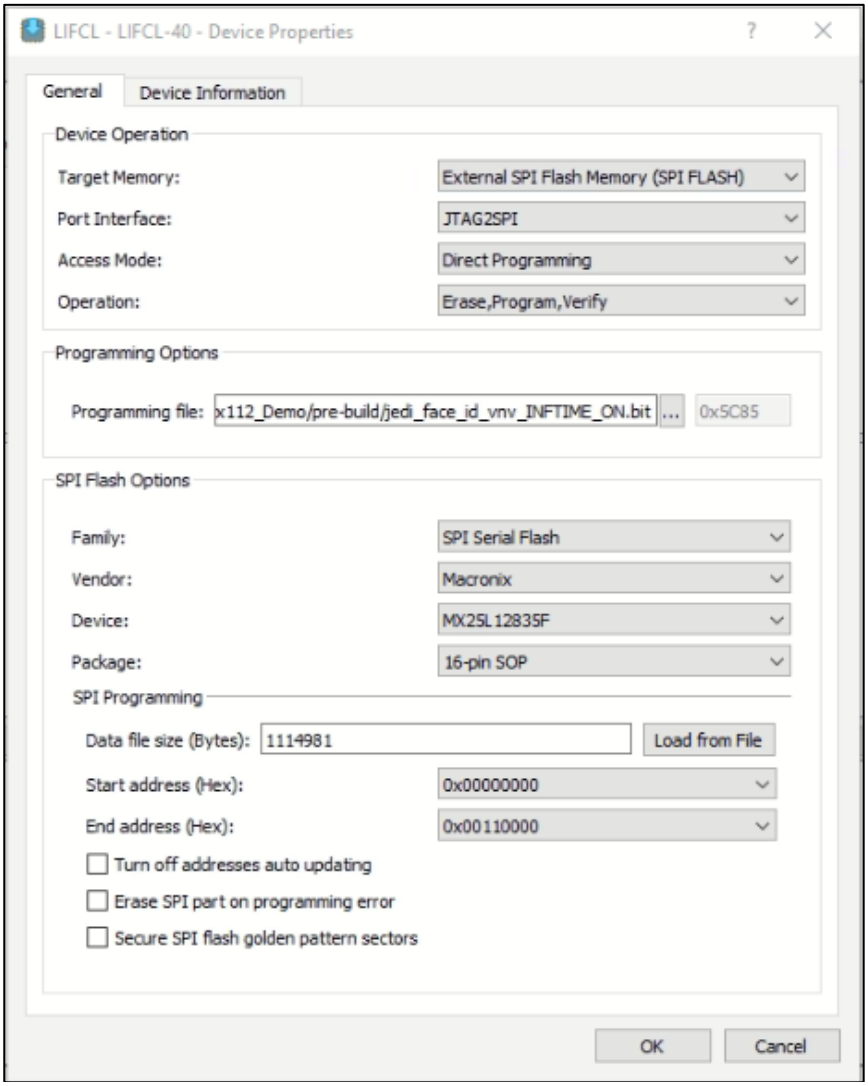

Figure 4.6. Lattice Radiant Programmer – Selecting Device Properties Options for CrossLink-NX Flashing

Notes:

- In **Programming file**, browse and select the CrossLink-NX Voice and Vision bit file (*.bit).
- Click **Load from File** to update the Data file size (bytes) value.
- Ensure that the following addresses are correct:
  - Start Address (Hex) – 0x00000000
  - End Address (Hex) – 0x00110000

4. Click **OK**.
5. Press and hold SW5 until you see the *Operation: successful* message in the Lattice Radiant log window.
6. Click the **Program** button to start the programming operation.

7. After successful programming, the **Output** console displays the result, as shown in Figure 4.7.

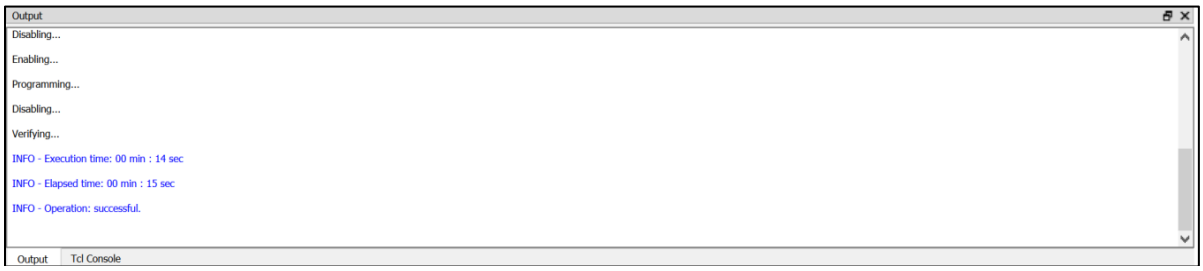

Figure 4.7. Lattice Radiant Programmer – Output Console

4.3.3. Programming sensAI Firmware Binary to the CrossLink-NX SPI Flash

To program the CrossLink-NX SPI flash:

1. Ensure that the CrossLink-NX device SRAM is erased by performing the steps in the Erasing the CrossLink-NX Voice and Vision SRAM Prior to Reprogramming section before flashing bitstream and sensAI firmware binary.
2. In the Lattice Radiant Programmer main interface, right-click the CrossLink-NX row and select **Device Properties**.
3. In the **Device Properties** dialog box, apply the settings as shown in Figure 4.8.

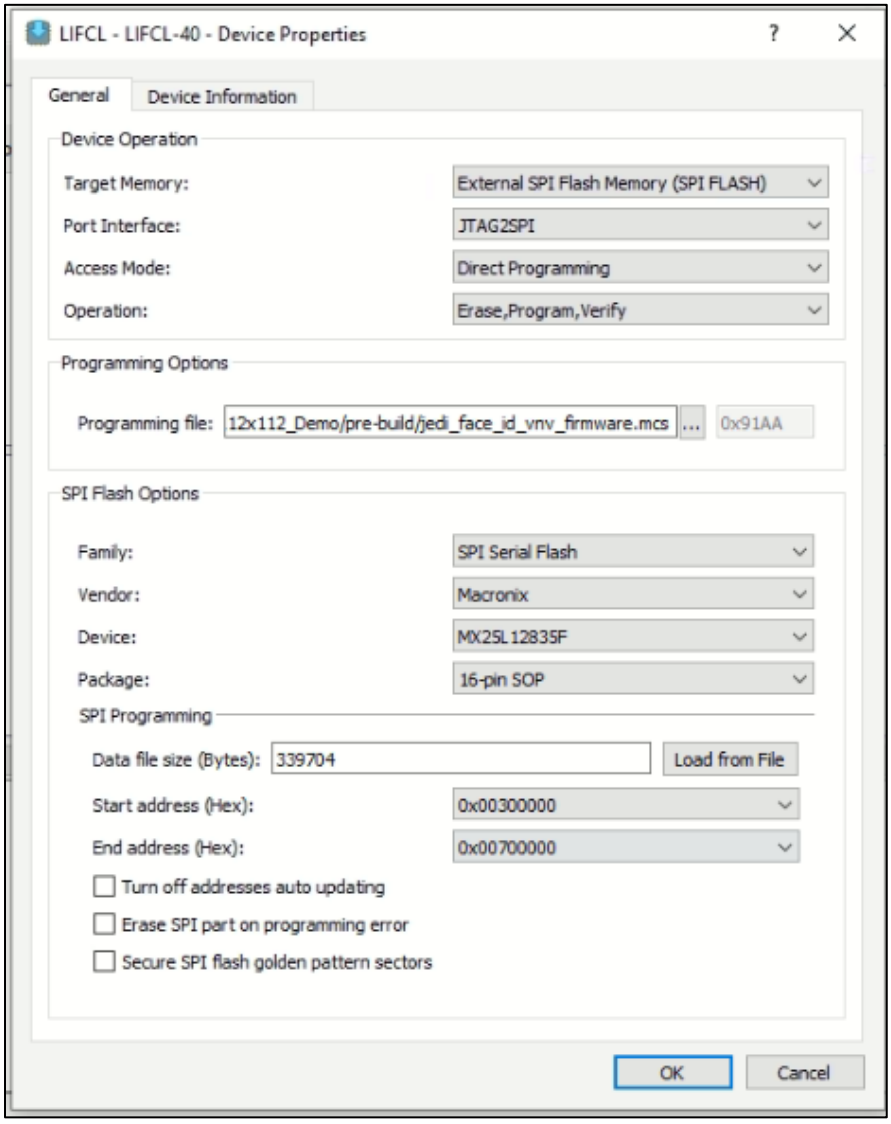

Figure 4.8. Lattice Radiant Programmer – Selecting Device Properties Options for CrossLink-NX Flashing

Notes:

- In **Programming file**, browse and select the CrossLink-NX sensAI firmware binary file after converting it to hex (*.mcs).
- Click **Load from File** to update the Data file size (bytes) value.
- Ensure that the following addresses are correct:
  - Start Address (Hex) – 0x00300000
  - End Address (Hex) – 0x00700000

4. Click **OK**.

5. Press and hold SW5 until you see the *Successful* message in the Lattice Radiant log window.
6. Click the **Program** button to start the programming operation.
7. After successful programming, the **Output** console displays the result, as shown in Figure 4.9.

Figure 4.9. Lattice Radiant Programmer – Output Console

5. Running the Demo

To run the demo:

1. Power on the VVML board.
2. Make sure that the position of DIP SWITCH0 is ON to set FX3 to boot from I^2C EEPROM.
3. Flash the .bit and .mcs files.
4. Connect the VVML board to the PC through the USB3 port.
5. Open the AMCap video display application and select the FX3 Device as source from under **Devices**.
6. The camera image with green guide lines enabled is displayed in the output Figure 5.1.
7. Keep the face close to the camera and inside the active area indicated by the guide lines.
8. Press the SW2 button on the CrossLink-NX VVML Board to register the face.
9. Press the SW3 button to erase the registered values and initialize the registration process from beginning.

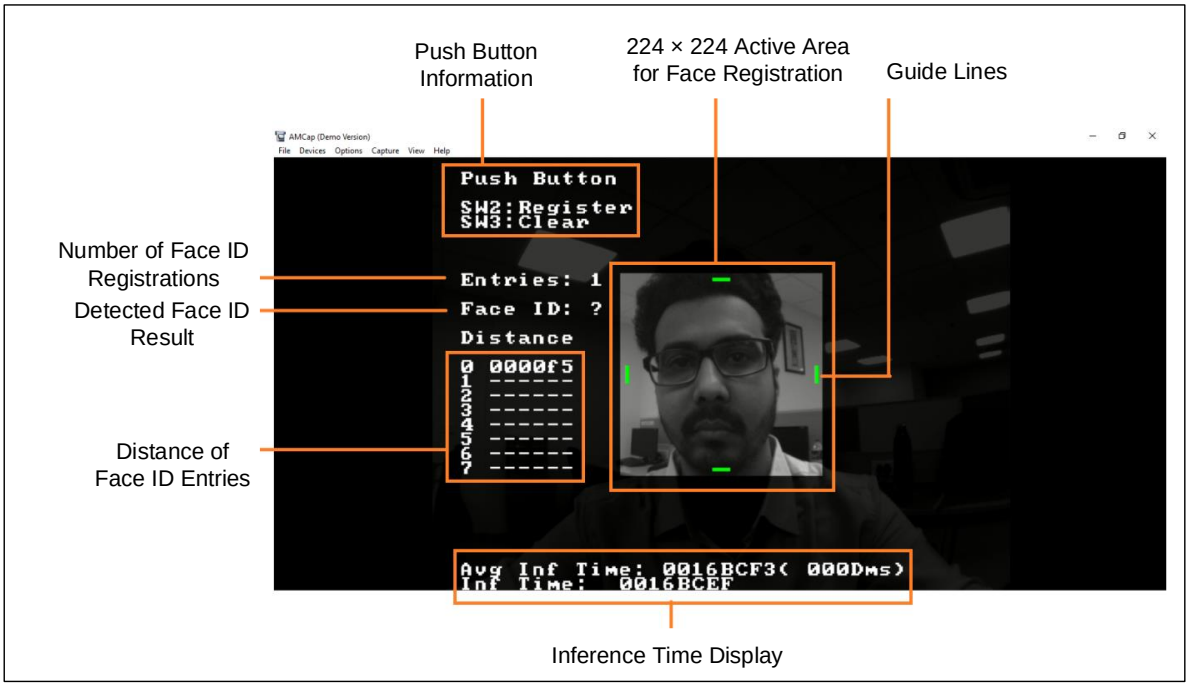

Figure 5.1. Demo Camera Image

If Inference time display is disabled, and .bit file is generated using *face_id_display.mem*. The text and values of Inference Time Display are not displayed in the output. The output shown in Figure 5.1 is with the .bit file having Inference time enabled using the *face_id_display_INF.mem* file.

5.1. Ideal Conditions for Testing the Demo

- Distance – The user's face should completely fit the guide lines.
- Lighting – Proper lighting is needed to efficiently run demo. Too low and direct light from a source may reduce the performance quality of the demo.

Technical Support Assistance

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

Revision History

Revision 1.0, November 2021

Section	Change Summary
All	Initial release

www.latticesemi.com