CrossLinkU-NX SoM for HMI Demonstration

Reference Design

FPGA-RD-02333-1.1

February 2026

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 6
1. Introduction 7
1.1. Quick Facts 8
1.1.1. Quick Start Guide 8
1.2. Features 9
2. Directory Structure and Files 10
3. Functional Description 11
3.1. MIPI Sensor Processing 11
3.2. AI Detection Module 11
3.2.1. AI Pipeline 12
3.2.2. AI Configuration 12
3.3. EVE SDK 12
3.4. Web Server 12
3.5. Remote Firmware Update 13
4. sensAI Edge Vision Engine Tool 14
4.1. Landing Page Screen 14
4.2. Metadata Screen 14
4.2.1. Metadata Presentation 15
4.2.2. Feature Enablement 16
4.2.3. Maximum IPS 17
4.2.4. Face ID on All Users 18
4.2.5. Register New Face ID 19
4.2.6. Clear Face ID Gallery 20
5. Running the HMI Demo Design on CrossLinkU-NX SoM HW Board 22
5.1. Requirements 22
5.1.1. Hardware 22
5.1.2. Software 22
5.2. Board Setup 23
5.2.1. CrossLinkU-NX SoM HW Board 23
5.2.2. Generic Carrier HW Board 24
5.2.3. Raspberry Pi CM5 25
5.2.4. IMX219 Camera Module (Narrow FOV – ArduCam B0191) 26
5.3. Board-to-Board Connections 27
5.4. Configuring the Hardware Jumper 28
5.5. RPi CM5 Serial Console Access 29
5.6. Connecting the RPi CM5’s Ethernet 31
5.6.1. Setting Up Local Network on Router 32
5.6.2. Setting Up Direct Connection to Host PC 33
5.6.3. Obtaining IP Address of Raspberry Pi CM5 38
5.7. Running EVE and Web Server 38
6. Running the Firmware Management Application 40
6.1. Updating the Firmware 40
6.2. Firmware Version 41
7. Debug Methodology 42
7.1. Powering Up the CrossLinkU-NX SoM 42
7.1.1. Troubleshooting Common Issues 43
7.1.2. Programming the Bit File 44
7.2. Verifying Ethernet Connection 46
7.3. Flashing the RPi CM5 47
References 50
Technical Support Assistance 51

Revision History 52

Figures

Figure 1.1. CrossLinkU-NX SoM Hardware Components for HMI Demo 7
Figure 1.2. System Overview of CrossLinkU-NX SoM for HMI Demo 7
Figure 3.1. Overview of HMI Demonstration Design Block Diagram 11
Figure 3.2. Web Server Sample 13
Figure 4.1. Landing Page Screen 14
Figure 4.2. Metadata Screen – Streaming Mode 15
Figure 4.3. Metadata Screen – Sensing Mode/ULP Sensing Mode 15
Figure 4.4. Metadata Presentation 16
Figure 4.5. Feature Enablement 16
Figure 4.6. Pipeline Control with Only Person Detection 17
Figure 4.7. Pipeline Control with Only Face Detection 17
Figure 4.8. Maximum IPS 18
Figure 4.9. Face ID on All Users setting in Pipeline Control 18
Figure 4.10. Face ID on All Users 18
Figure 4.11. Face ID on Main User Only 19
Figure 4.12. Register New Face ID 19
Figure 4.13. Registered Face ID 20
Figure 4.14. Face ID – Requirement Unmet 20
Figure 4.15. Clear Face ID Gallery 20
Figure 4.16. Face ID with No Gallery Shown After Clear 21
Figure 5.1. CrossLinkU-NX SoM HW Board (Top View) 23
Figure 5.2. CrossLinkU-NX SoM HW Board (Bottom View) 24
Figure 5.3. Generic Carrier HW Board (Top View) 25
Figure 5.4. Generic Carrier HW Board (Bottom View) 25
Figure 5.5. Raspberry Pi CM5 (Top View) 26
Figure 5.6. Raspberry Pi CM5 (Bottom View) 26
Figure 5.7. IMX219 Camera Module (Top View) 26
Figure 5.8. IMX219 Camera Module (Bottom View) 27
Figure 5.9. Stacking Orientation Arrangement 27
Figure 5.10. Board-to-Board Connections 28
Figure 5.11. Board Connections with Camera FFC Cable 28
Figure 5.12. Hardware Connections for UART 29
Figure 5.13. Hardware Setup for Serial Console Access 29
Figure 5.14. Available COM Ports in Device Manager 30
Figure 5.15. PuTTY Configuration to Open COM Port 30
Figure 5.16. RPi CM5 Login Screen 31
Figure 5.17. Serial Terminal Prompt 31
Figure 5.18. CrossLinkU-NX SoM with Ethernet Connection 32
Figure 5.19. Control Panel 33
Figure 5.20. Network and Internet 34
Figure 5.21. Network and Sharing Center Page 34
Figure 5.22. Network Connections 35
Figure 5.23. Changing Internet Protocol Version 4(TCP/IPv4) Settings 35
Figure 5.24. Static IP Settings 36
Figure 5.25. Network Adapter Properties 37
Figure 6.1. Hardware Connections for Firmware Update 40
Figure 6.2. Firmware Update Success Message 41
Figure 6.3. Firmware Versions 41
Figure 7.1. Debug Methodology Flow Chart 42

Figure 7.2. LED Position on CrossLinkU-NX SoM 43
Figure 7.3. Hardware Setup for Bit File Programming..44
Figure 7.4. Radiant Programmer – Getting Started Window..44
Figure 7.5. Radiant Programmer Detect Cable Settings 45
Figure 7.6 Selecting Bitstream File to Program 45
Figure 7.7 Program Device Toolbar Icon..46
Figure 7.8. Message on Successful Programming...46
Figure 7.9. Verify Raspberry Pi CM5 Ethernet Connectivity 47
Figure 7.10. Hardware Connection for nRPIBOOT...47
Figure 7.11. Hardware Setup for Flashing RPi CM5 OS..48
Figure 7.12. Configuring RPi CM5 as Mass Storage Device 48
Figure 7.13. Pop Up to Insert Disk 48

Tables

Table 1.1. Summary of the Demonstration Files 8
Table 2.1. Folder Description..10
Table 3.1. MIPI Sensor Processing Features 11
Table 3.2. Supported ML Models...12
Table 3.3. Operational Modes 13
Table 5.1. Available Hardware in the CrossLinkU-NX SoM Demo Kit (OPN: LIFCL-SOM-EVN) 22
Table 7.1. LED Status Upon Board Power Up 43

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviations	Definition
AI	Artificial Intelligence
CM5	Compute Module 5
CNN	Convolutional Neural Network
CPU	Central Processing Unit
Demo Design	Demonstration Design
ECDSA	Elliptic Curve Digital Signature Algorithm
EVE	Edge Vision Engine
FFC	Flat Flexible Cable
FM	Firmware Management
FOV	Field of View
FW	Firmware
GPIO	General Purpose Input/Output
HMI	Human Machine Interface
IPS	AI Inference Per Second
JTAG	Joint Test Action Group
micro-ISP	Micro Image Signal Processing
OPN	Ordering Part Number
OS	Operating System
PC	Personal Computer
RPi	Raspberry Pi
SDK	Software Development Kit
SoM	System on Module
UI	User Interface
ULP	Ultra-Low Power
WiFi	Wireless Fidelity

1. Introduction

This user guide introduces the CrossLinkU™-NX System-on-Module (SoM) HMI Demonstration Design, purpose-built for Human-Machine Interaction (HMI) applications. The design enables AI-driven interaction capabilities, including person detection and face recognition enabling intuitive and intelligent user experiences. Optimized for edge device processing, the solution supports ultra-low power mode, while adhering to industry standards for embedded vision and AI systems.

If you plan to use the CrossLinkU™-NX SOM to enable HMI-based interactions in a custom application, refer to the Communication Protocols and the EVE API Guide for the CrossLinkU-NX SoM (FPGA-UG-02250) document, which describes several methods for reading the metadata, allowing you to choose the approach that best meets your objectives and requirements.

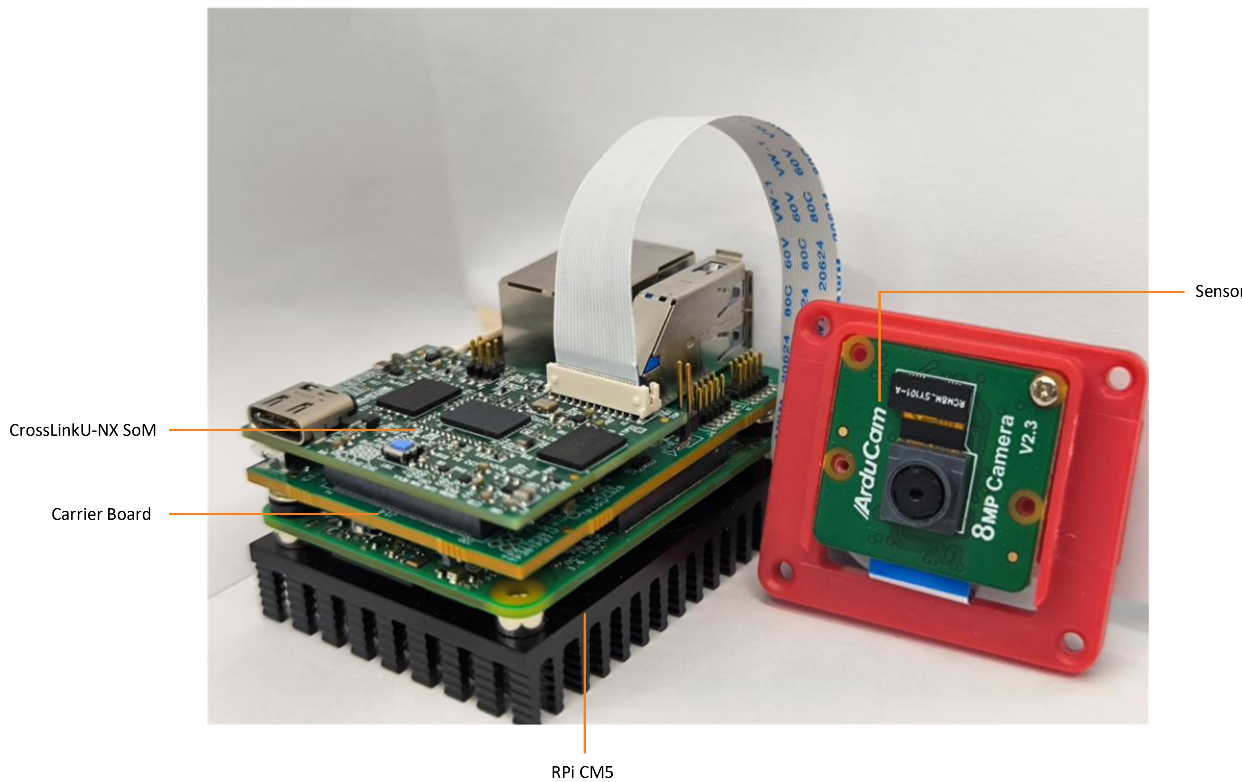

Figure 1.1. CrossLinkU-NX SoM Hardware Components for HMI Demo

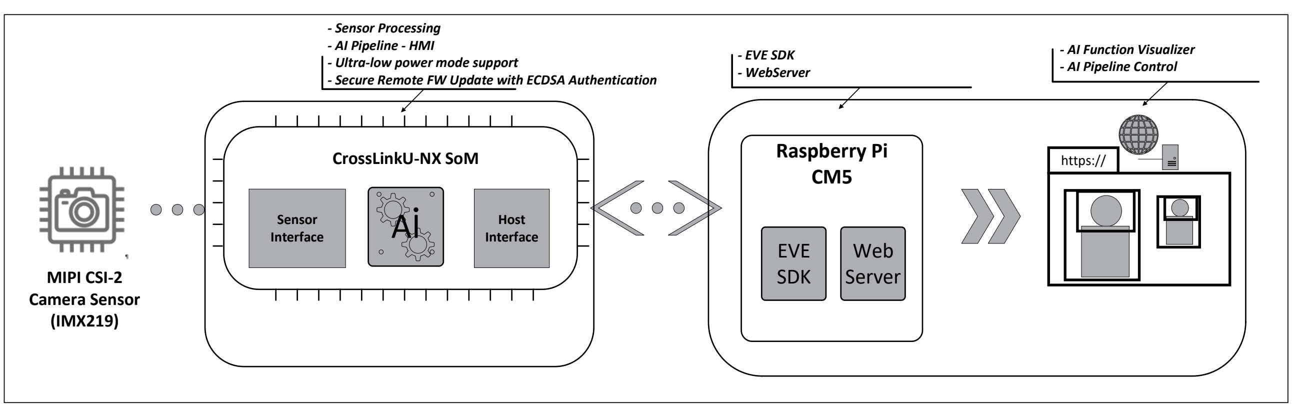

Figure 1.2. System Overview of CrossLinkU-NX SoM for HMI Demo

The key components of the CrossLinkU-NX SoM HMI Solution are:

- CrossLinkU-NX System on Module HW board (LIFCL-33U-SOM-EVN)
  - Powered by the Lattice CrossLinkU-NX with HMI Demo Firmware.
  - Performs sensor MIPI streaming to host system
  - Integrated AI pipeline with CNN Plus IP for HMI detection
  - Implements Low-power AI processing leverage CrossLinkU-NX AON controller
- Generic Carrier HW Board (LF-GEN-CR-PCBA)
  - Interfacing between CrossLinkU-NX SoM HW Board and Raspberry Pi (RPi) CM5
  - On-board FTDI Controller for:
    - PortA: CrossLinkU-NX SoM and CrossLinkU-NX JTAG for Configuration
    - PortB: Reserved
    - PortC: Reserved
    - PortD: CM5's UART Serial Console Access (Setup)
  - Bridges CM5 USB-C port, USB-Type A port and Ethernet Port
- Raspberry Pi Compute Module 5
  - Equipped with USB, Ethernet, and GPIO interfaces
  - Acts as a host to handle AI metadata processing with EVE SDK and visualization dashboard with the web server
- Sensor
  - Targeting MIPI CSI-2 camera sensor IMX219
  - ArduCam B0191, Narrow FOV, Coverage 62.2°(H) × 48.8°(V)

1.1. Quick Facts

Download the HMI demonstration files from the HMI Demonstration web page.

Table 1.1. Summary of the Demonstration Files

General	Target Devices	CrossLinkU-NX (LIFCL-33U-FCCSP104)
	Source format	Bitstream
Software Requirements	Software tool and version	• Lattice Radiant™ Programmer 2025.1 or later • Raspberry Pi Imager v1.9.6 • Raspberry rpiboot
	IP Version (if applicable)	Not applicable
	Host Software	• Raspberry Pi OS – Bookworm[1] • Lattice EVE SDK • Lattice Web Server • PuTTY or equivalent (for example, Tera Term)
Hardware Requirements	Demo Kit	CrossLinkU-NX SoM Demo Kit (LIFCL-33U-SOM-EVN)
	Host	Raspberry Pi CM5

Note:

1. Lattice Custom RPi OS for CrossLinkU-NX SoM HMI Demo. OS update is disabled by default.

1.1.1. Quick Start Guide

Refer to the Running the HMI Demo Design on CrossLinkU-NX SoM HW Board section for instructions on setting up and running HMI demonstration on the Raspberry Pi CM5 with CrossLinkU-NX SoM.

1.2. Features

The CrossLinkU-NX SoM Demo Design – HMI Demonstration offers the following key capabilities:

- Sensor Interface
    - Support 2-lanes MIPI CSI-2 Camera Sensor (IMX219), Narrow FOV
    - Support MIPI Streaming to Host for Camera Preview
    - Support MIPI for AI Processing
- AI Pipeline
    - Integrated AI pipeline utilizing CNN Plus IP for edge AI inference
    - Support AI Features such as Person Detection, Face Detection, Face Landmark Detection, and Face Identification
    - Metadata output through I2C with interrupt event signaling
    - Optimized for Low-power architecture with CrossLinkU-NX AON support
- Host
    - Runs on Raspberry Pi OS – Bookworm
    - Enables real-time AI visualization using the Lattice EVE SDK, with web server interface support
- Firmware Update
    - Support remote firmware update through Lattice Firmware Management Application
    - Secure firmware update mechanism with ECDSA authentication

2. Directory Structure and Files

Table 2.1 illustrates the folder description of the CrossLinkU-NX SoM HMI Demo Package. Comprehensive information is available in the GitHub README file, which is linked on the HMI Demonstration web page. The file names may vary across different packages. Refer to the *GitHub README* file for the download instructions.

Table 2.1. Folder Description

Folder/File	Description
CLNX_SPI_IMAGE[1]	This folder contains the 64 MB CrossLinkU-NX SoM SPI Flash Binary.
EVE_WEBSERVER[1]	This folder contains the installer for EVE and Webserver.
FW_UPDATE_TOOL[1]	This folder contains the CrossLinkU-NX SoM Firmware Management application.
RESOURCES[1]	This folder contains the resource files and scripts.
RESOURCES/RPI_SCRIPTS[1]	This folder contains the scripts used for the demo and programmer scripts.
RPI_DRIVERS[1]	This folder contains the Lattice drivers and overlays used for the demo.
RPI_OS	This folder contains the Raspberry Pi OS image.[2]

Notes:

1. These contents are preinstalled into the Raspberry Pi OS Image.
2. Lattice Custom RPi OS for CrossLinkU-NX SoM HMI Demo. OS update is disabled by default.

3. Functional Description

This section describes the architecture of the HMI demonstration with Lattice CrossLinkU-NX SoM. The system architecture of the demonstration design involves several key subsystems working together seamlessly. There are five main subsystems involved in the design:

- MIPI Sensor Processing
- AI Detection
- EVE SDK
- Web Server
- FW Update

Figure 3.1 shows the overview of the HMI demonstration design block diagram and five main subsystems.

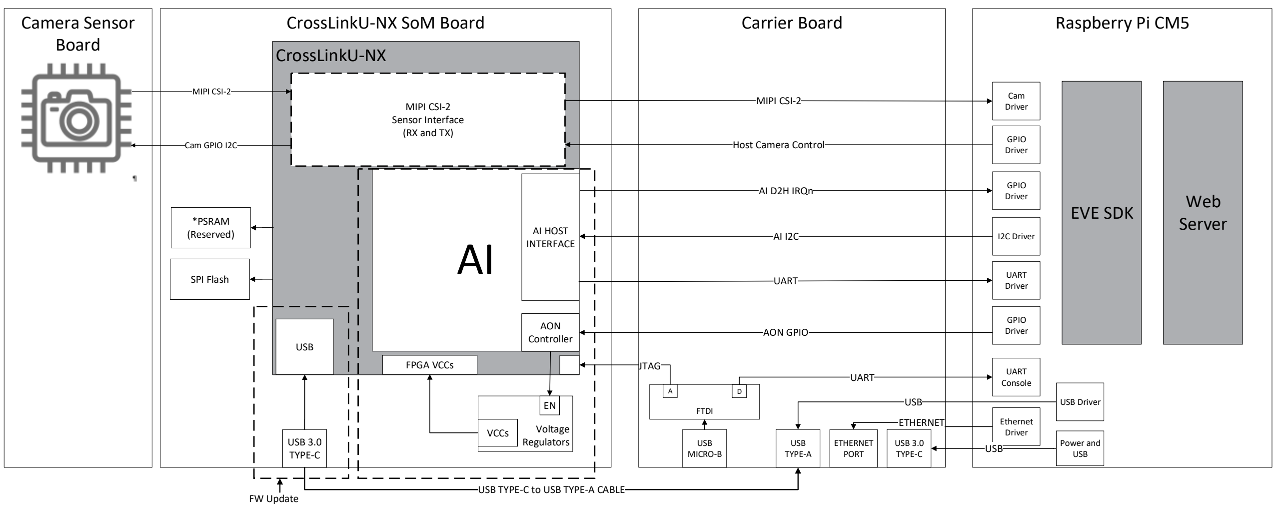

Figure 3.1. Overview of HMI Demonstration Design Block Diagram

3.1. MIPI Sensor Processing

The design implements 2-lane MIPI CSI-2 sensor processing. In the *Streaming* mode, the design supports MIPI passthrough, enabling uninterrupted video transmission from the camera to the host system. Concurrently, the camera stream from MIPI RX is fed to AI processing pipeline for HMI AI Inferencing.

Table 3.1. MIPI Sensor Processing Features

Feature	Specification
Camera Sensor Module	Narrow FOV: IMX219 – ArduCam B0191
MIPI	RX and TX: CSI-2, 2-Lanes, RAW10, up to 960 Mbps per lane.

3.2. AI Detection Module

The AI pipeline leverages CNN Plus IP to deliver accelerated inference for edge AI applications. The supported models include Person Detection, Face Detection, Face Validation, and Face Identification. The pipeline incorporates pre-processing (micro-ISP), Machine Learning execution, and post-processing stages. AI inference results are transmitted as metadata over AI I2C, accompanied with interrupt signal to Host for event notification. Metadata includes bounding box coordinates, distance from the camera, head pose orientation and others. The AI pipeline supports ULP (Ultra-Low Power) mode which utilizes the CrossLinkU-NX AON Controller to further optimize the power consumption.

Table 3.2. Supported ML Models

Model	Function
Person Detection	Detects human bodies
Face Detection	Detects the human face
Face Validation	Detects and validates face landmarks
Face ID	Face identification

3.2.1. AI Pipeline

The AI pipeline consists of an infinite loop of inferences and a metadata output. Every loop in the basic configuration performs the following:

- Run Person Detection
- Run Face Detection
- Run Face Validation
- Run Face ID
- Output Metadata

3.2.2. AI Configuration

The AI pipeline can be configured from the Host at runtime. For each supported model, it can configure:

- Enablement
    - By default, each AI model is enabled.
    - If a model is enabled, its output is present in the metadata sent to the host at the end of the loop.
- Maximum Inference Per Second (IPS)
    - By default, the Maximum IPS of each model is *MAX* (value of 0). It runs as fast as possible with no restriction.
    - If it is set to something different than 0 for a specific model, the pipeline may not run the model on every loop to prevent exceeding the Maximum IPS value.
    - The pipeline can pause at the end of a loop to not output faster than the highest Maximum IPS out of all the models.

In ULP mode, the AI model control and maximum IPS option are disabled. The pipeline assumes full control of the IPS. This mode is ideal for showcasing energy-efficient operation.

3.3. EVE SDK

The EVE SDK is a software library that parses HMI AI metadata and works alongside the CrossLinkU-NX SoM to provide real-time insights and structured visualizations. It processes metadata from the AI pipeline and formats it for easy visualization and analysis. When an AI interrupt occurs, the EVE SDK signals that new data is available. This data is then parsed into a defined structure, including bounding boxes, distances, head pose orientation, and other details that overlay the current image.

3.4. Web Server

The Raspberry Pi CM5 hosts a web server that provides an interactive dashboard for monitoring and control. Through this dashboard, users can visualize AI outputs, review system performance metrics, and configure the AI pipeline at runtime to meet application requirements.

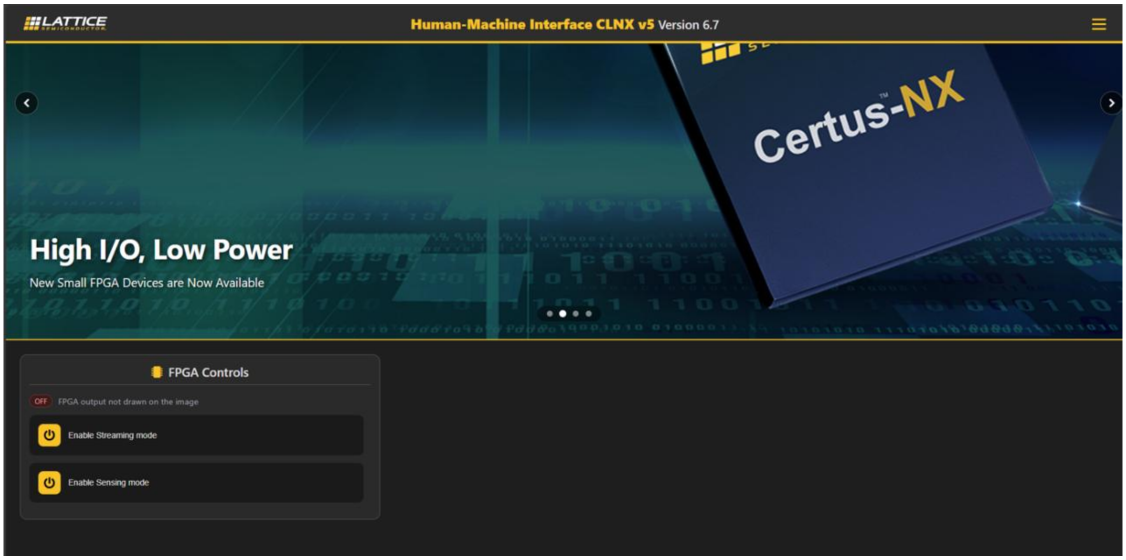

Figure 3.2. Web Server Sample

The HMI demo design supports three operational modes (Streaming, Sensing, and ULP Sensing) as described in Table 3.3. These modes can be configurable through EVE SDK and Web Server.

Table 3.3. Operational Modes

Mode	Description
Streaming	Continuous MIPI camera streaming to host while concurrently running AI models.
Sensing	AI detection only, without video passthrough.
ULP Sensing	Ultra-Low Power Always On at ~15mW[1]

Note:

1. During ULP Standby Operation.

3.5. Remote Firmware Update

The HMI demo design supports secure and authenticated firmware updates through USB, ensuring system integrity and resilience against unauthorized tampering. When a firmware update initiated, the system transitions into Firmware Management mode, receives new firmware from USB interface to program into SPI flash and authenticates them using ECDSA signatures. Once the process is completed, the system jumps to verified binaries. During the update process:

- Camera streaming and AI operations are temporarily paused.
- These functions resume only after the firmware update is successfully completed.

A USB cable is required to connect the CrossLinkU-NX SoM HW board USB-C port to the generic carrier HW board's USB Type-A port. This connection is essential for performing firmware updates. The update process uses the Lattice Firmware Management Application. Refer to the Running the Firmware Management Application section for further instructions.

4. sensAI Edge Vision Engine Tool

The Web Server hosts a graphical web application built on the EVE SDK. This section provides an overview of each screen in the application and explains how to navigate and use its features. The Web Server provides a web application that wraps the EVE SDK with a graphical interface. The following sections describe each screen and how to use them.

4.1. Landing Page Screen

This is the initial screen displayed when the web server starts and the address is accessed through a browser. From this screen, the *Streaming Mode* can be selected to display the camera's live feed, or the *Sensing Mode* to display the metadata only.

Notes:

1. Terminate the Web Server: After completing the operation, terminate the web server by pressing CTRL+C in the terminal where the application was launched.
2. Switching Modes: To switch between streaming and sensing modes, first terminate the web server. Then, relaunch the web server to return to the landing page.

Figure 4.1 shows the first screen that appears after launching the web server.

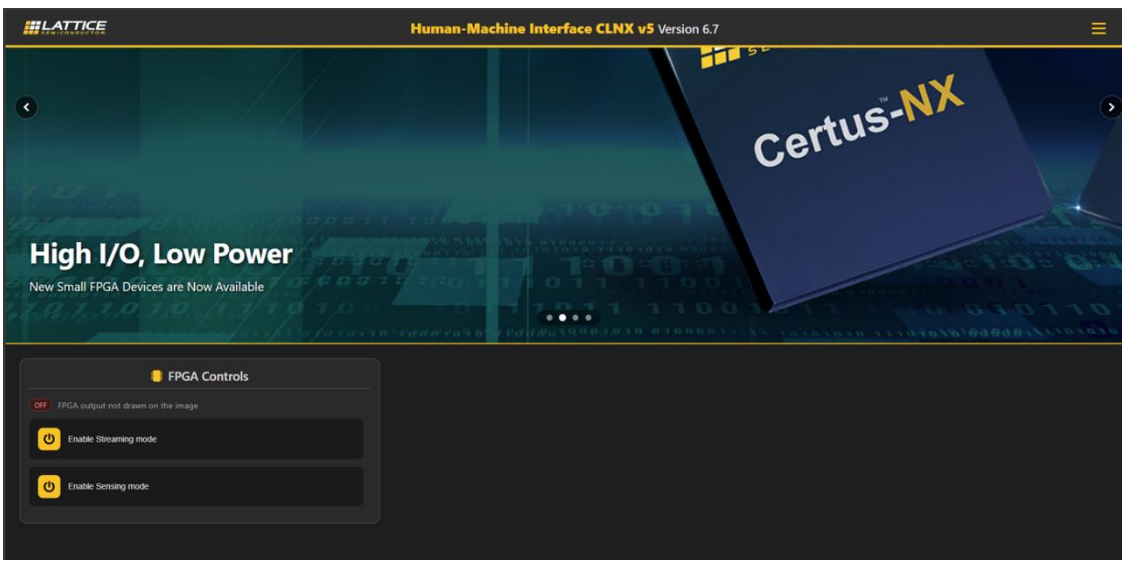

Figure 4.1. Landing Page Screen

4.2. Metadata Screen

This screen overlays metadata from the AI pipeline on the Live Video panels.

The Live Video panel updates based on the mode selected in the Initialization Screen:

- Streaming Mode – Displays live video with a metadata overlay, intended for applications that require video streaming and AI inferencing.

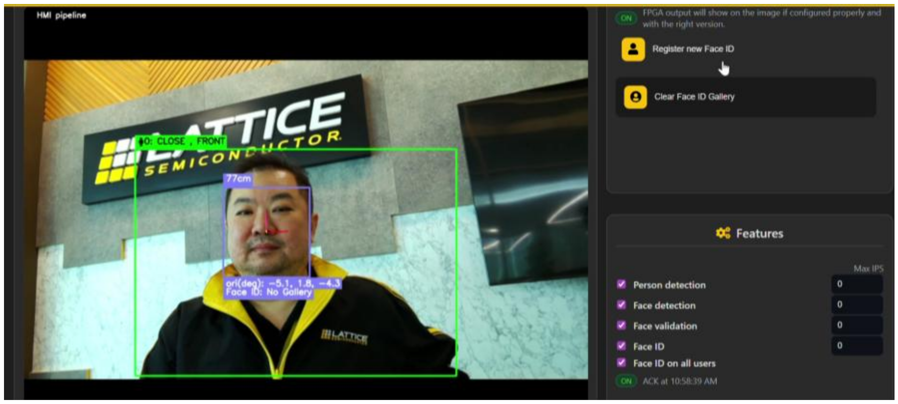

Figure 4.2. Metadata Screen – Streaming Mode

- Sensing Mode – The interface renders AI metadata over a neutral gray background. This mode is designed for applications that do not require video streaming and prioritize low-power operation. Additionally, it enhances privacy by ensuring the video feed is not transmitted to the host.
- ULP Sensing Mode – Within *sensing* mode, you can switch to Ultra-Low Power (ULP) mode using the Enable/Disable ULP Mode button. Once ULP mode is activated, all feature controls are disabled, as this mode is optimized to maintain the expected low-power operation.

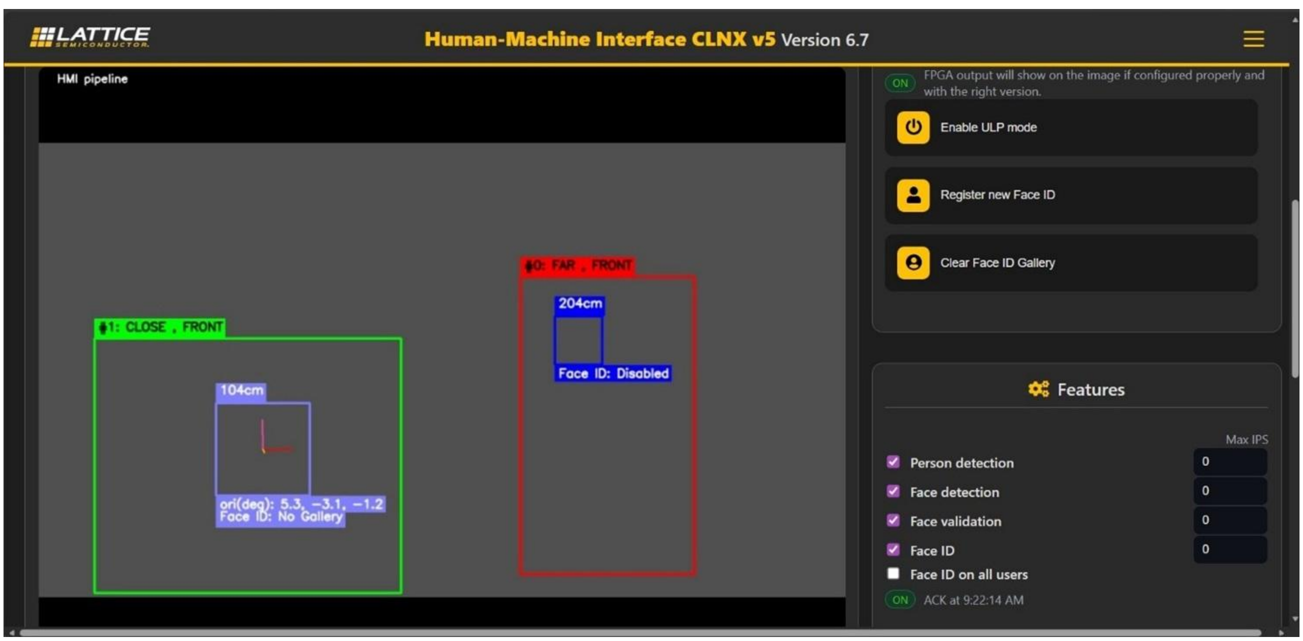

Figure 4.3. Metadata Screen – Sensing Mode/ULP Sensing Mode

4.2.1. Metadata Presentation

The metadata is displayed using colored bounding boxes combined with descriptive text. There are two types of bounding boxes used: Person Detection and Face Detection.

- Person Detection Bounding Box
  - Colour Codes:
    - *Green* for main user
    - *Red* for secondary person
  - Top Label Text: Shows *#INDEX: DEPTH, POSE*
    - INDEX: Person Index. Ranging from *0* to *4*.
    - DEPTH: Relative distance of the person from the camera, categorized as *CLOSE* or *FAR*.

   - POSE: Orientation of the person's body, with possible values: *FRONT* or *NON FRONTAL.*
- Face Detection Bounding Box
  - Colour Codes:
    - *Light Blue* for the Main User.
    - *Dark Blue* for Secondary Persons.
  - Top Label Text: Display distance from Camera in cm.
  - Bottom Label Text:
    - First Row – Shows *Yaw, Pitch, Roll* in degree.
    - Second Row – Indicates Face ID status, which may include: Unregistered, FID Number, Requirement unmet, Disabled, No Gallery.

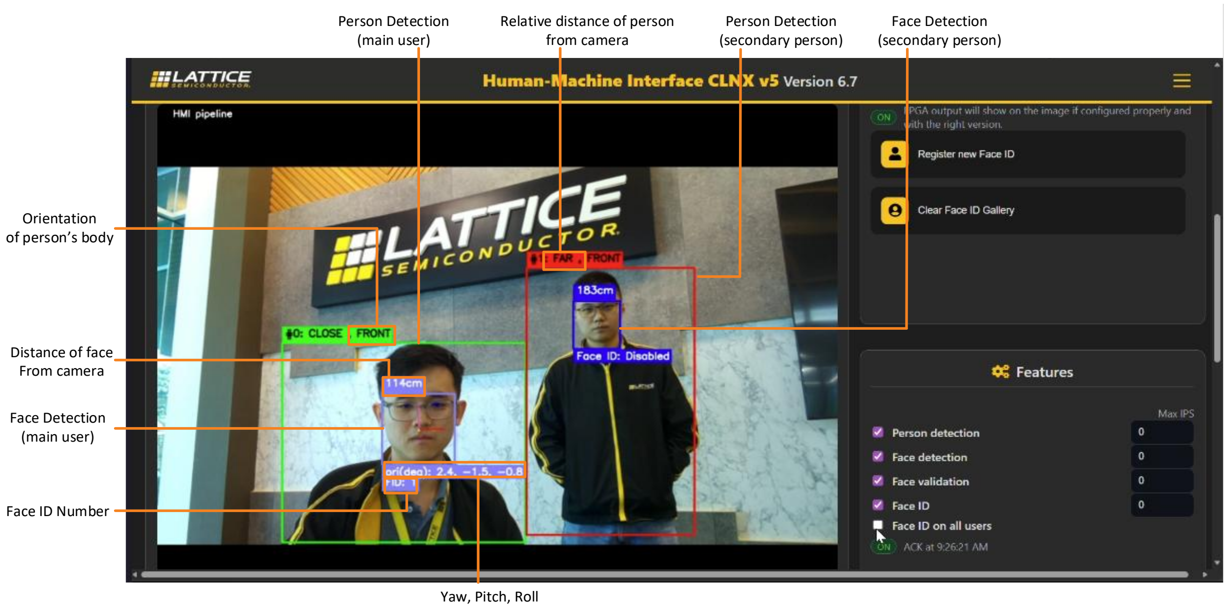

Figure 4.4. Metadata Presentation

4.2.2. Feature Enablement

The checkboxes dictate if a feature needs to run. By default, all the features are running. If a feature depends on other features to run, it silently enables those features without displaying it in the user interface.

- Face Validation depends on Face Detection
- Face ID depends on Face Detection and Face Validation

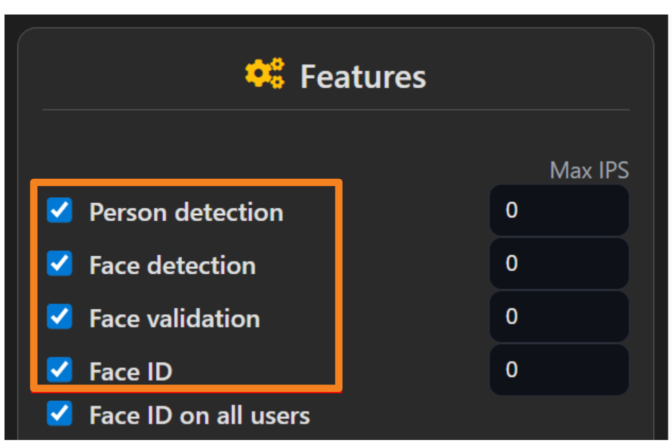

Figure 4.5. Feature Enablement

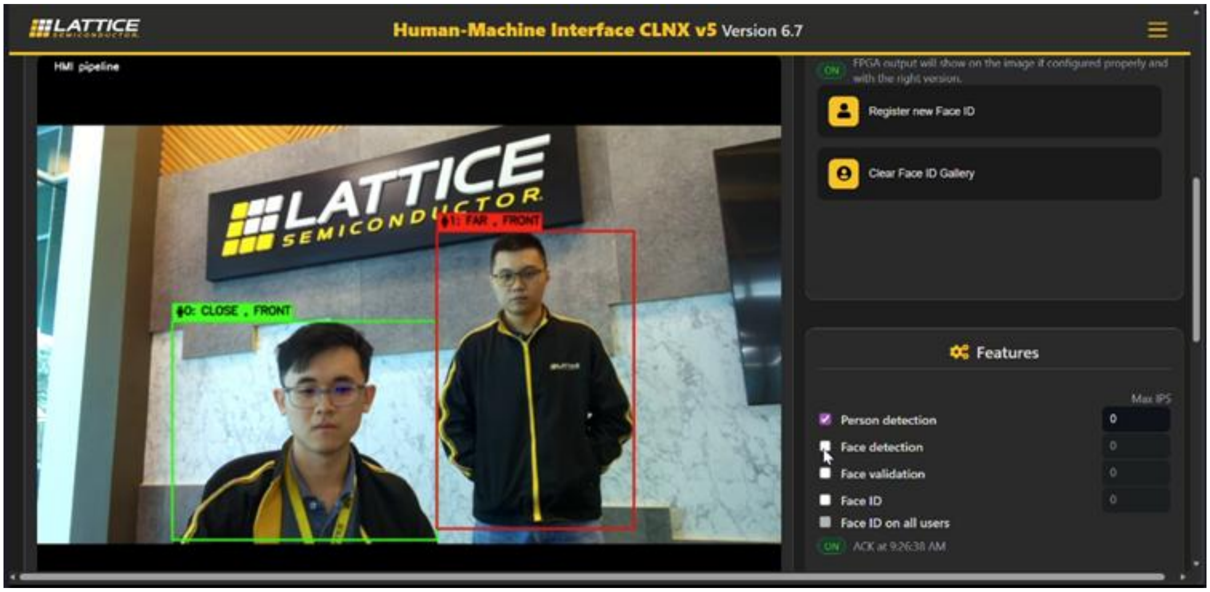

Figure 4.6. Pipeline Control with Only Person Detection

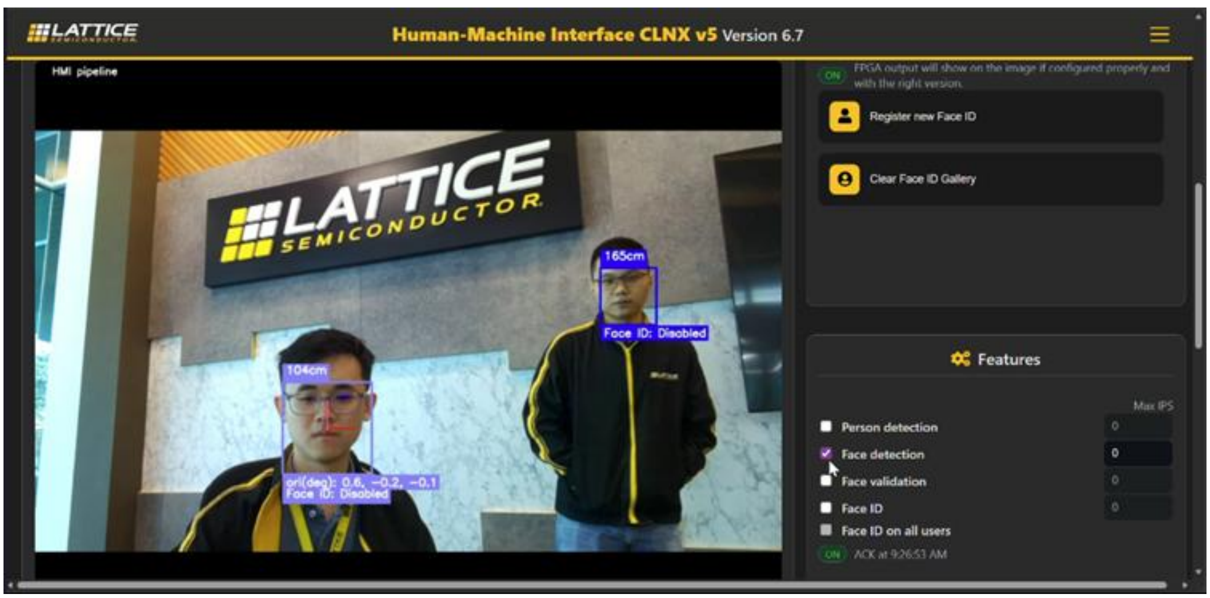

Figure 4.7. Pipeline Control with Only Face Detection

4.2.3. Maximum IPS

The *Max IPS* setting sets the maximum number of times the feature can run per second. The AI Pipeline section describes the pipeline operation.

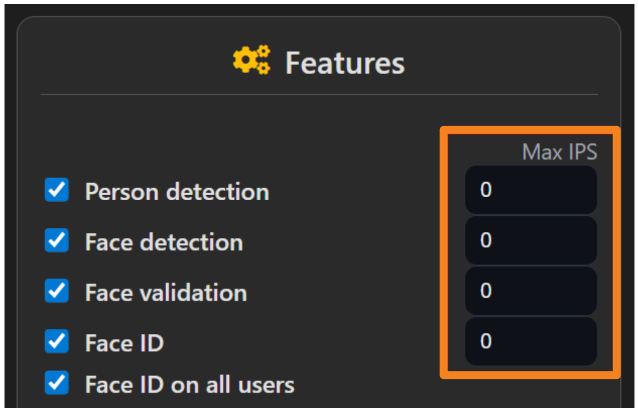

Figure 4.8. Maximum IPS

4.2.4. Face ID on All Users

The *Face ID on all users* setting, when enabled, runs Face ID checks on everyone in Face Detection range and displays the results under all face boxes as shown in Figure 4.10. If it is unchecked, the Face ID only runs on the main user as shown in Figure 4.11.

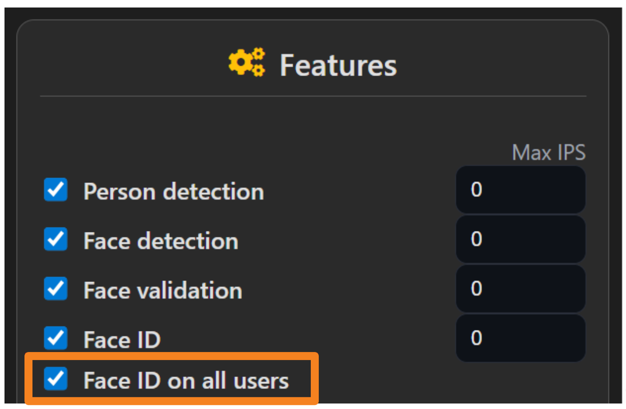

Figure 4.9. Face ID on All Users setting in Pipeline Control

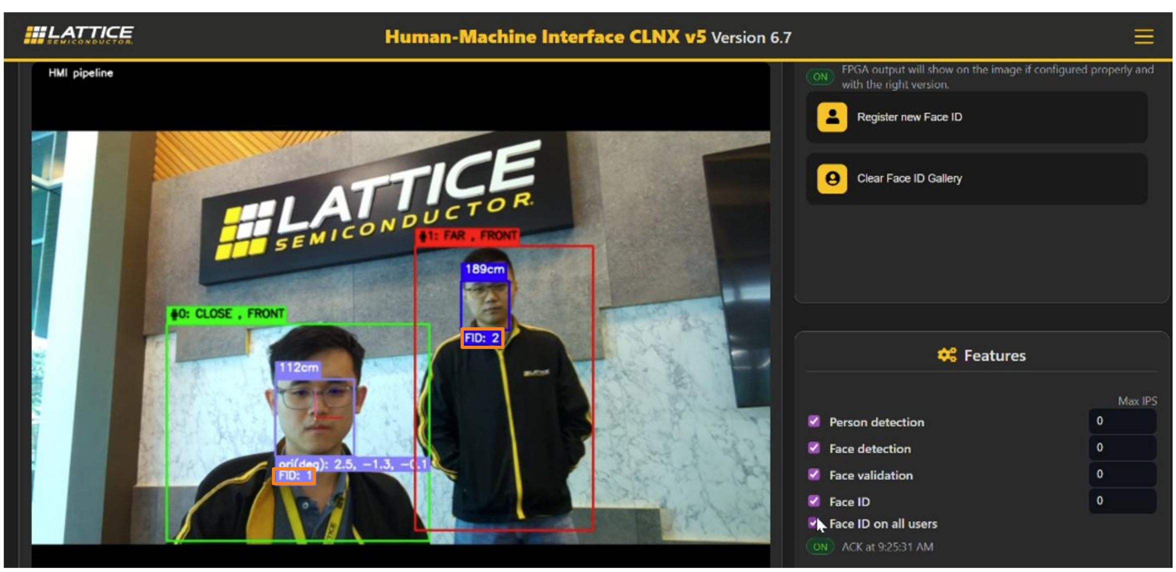

Figure 4.10. Face ID on All Users

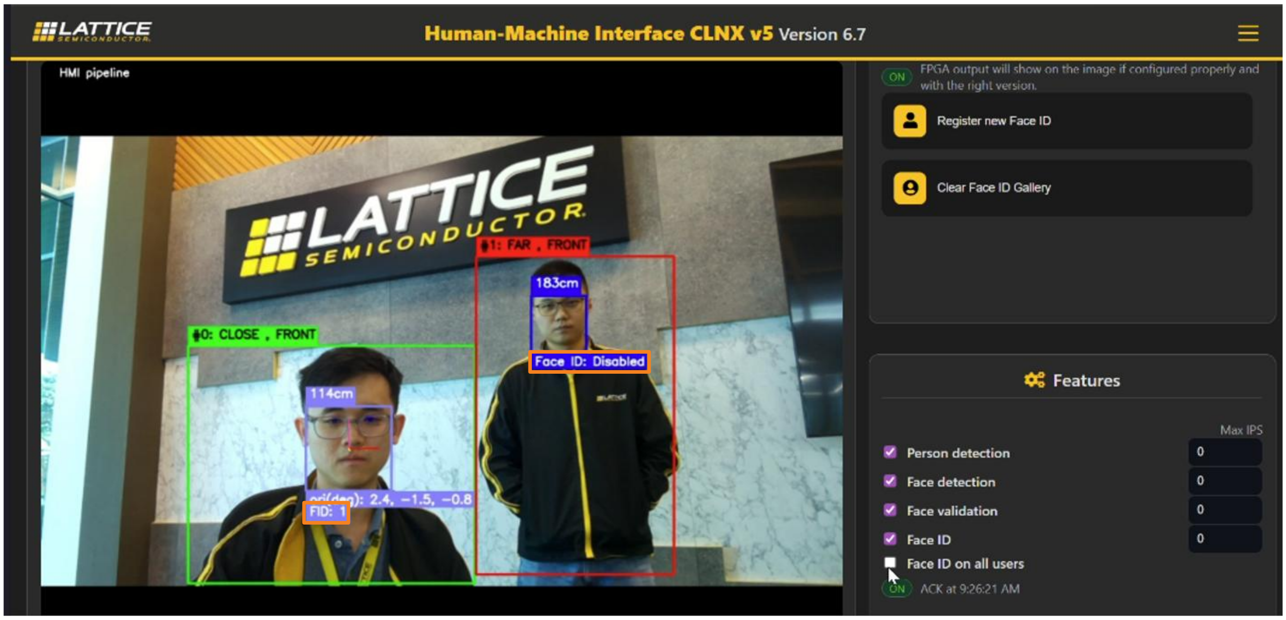

Figure 4.11. Face ID on Main User Only

4.2.5. Register New Face ID

The *Register new Face ID* button triggers a registration on the primary user denoted by the light blue face box. Subsequent Face ID checks return the user index in the gallery as shown in Figure 4.13.

Registration only works if the primary user is looking directly at the camera. If the requirement is not met, the registration is attempted until it meets the following conditions:

- Pitch > -20° and Pitch < 20°
- Yaw > -15° and Yaw < 15°
- Roll > -15° and Roll < 15°

In addition, the *Requirement Unmet* is labeled on the metadata screen shown in Figure 4.14.

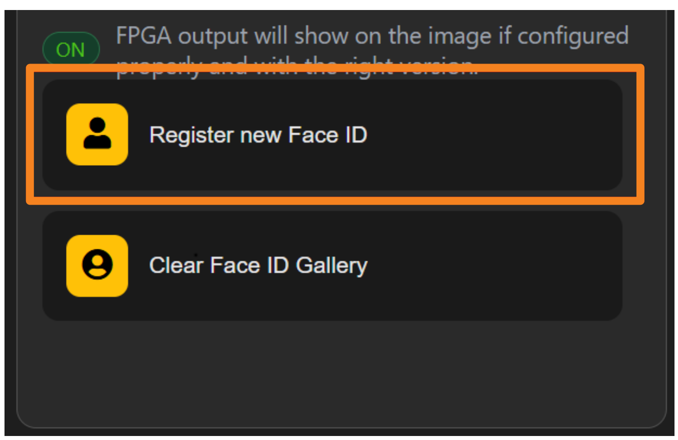

Figure 4.12. Register New Face ID

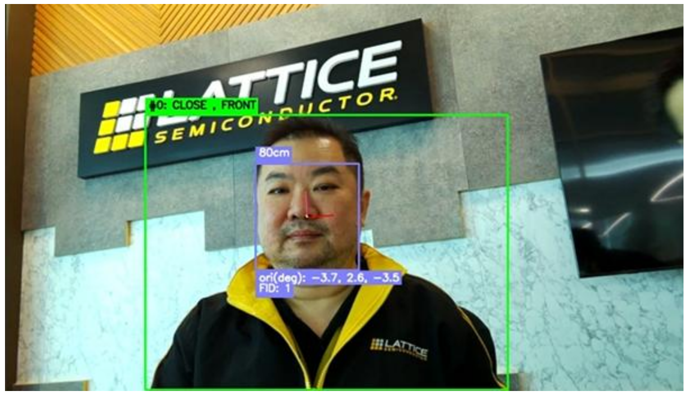

Figure 4.13. Registered Face ID

Figure 4.14. Face ID – Requirement Unmet

4.2.6. Clear Face ID Gallery

The *Clear Face ID Gallery* button removes everyone that registered from the gallery. After this, the result of Face ID must be *No gallery*.

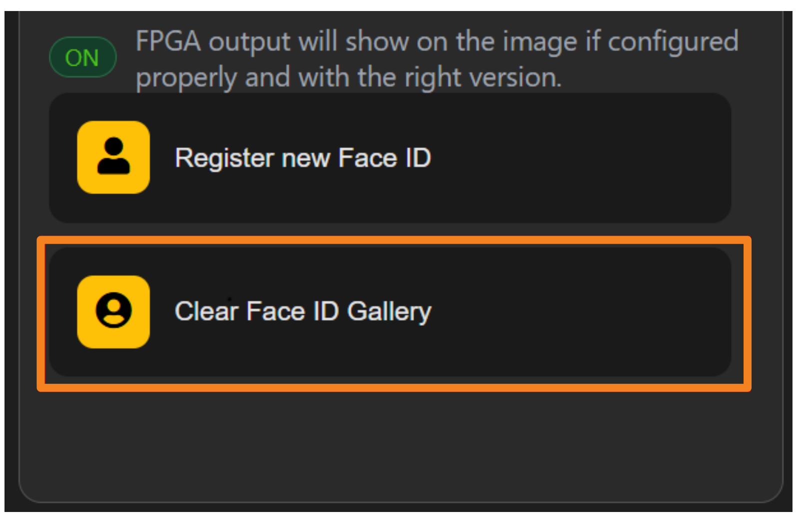

Figure 4.15. Clear Face ID Gallery

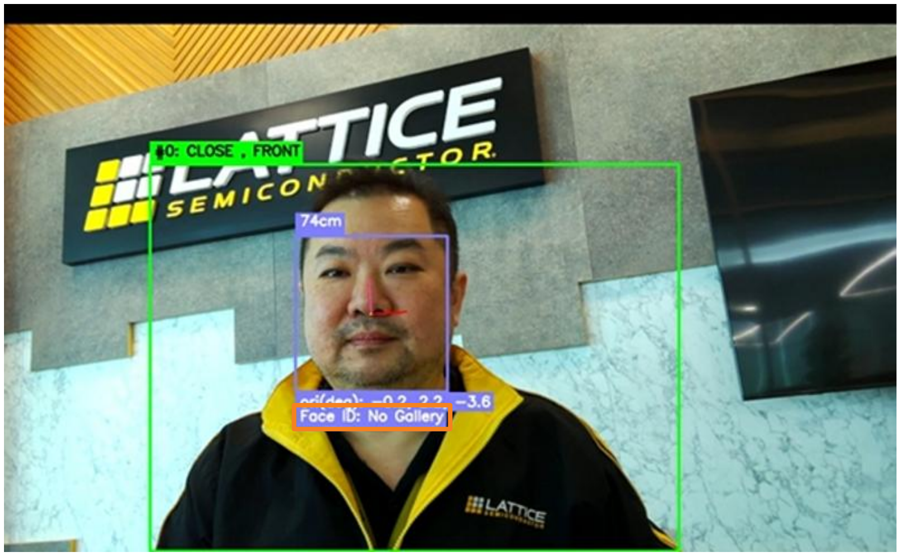

Figure 4.16. Face ID with No Gallery Shown After Clear

5. Running the HMI Demo Design on CrossLinkU-NX SoM HW Board

5.1. Requirements

The following lists the requirements for the design demonstration.

5.1.1. Hardware

Table 5.1. Available Hardware in the CrossLinkU-NX SoM Demo Kit (OPN: LIFCL-SOM-EVN)

Quantity	Description	Part Number/Manufacturer
1	CrossLinkU-NX SoM HW Board	PN: LIFCL-SOM-EVN Rev B, MFG: Lattice Semiconductor
1	Generic Carrier HW Board	PN: LF-GEN-CR-PCBA Rev A, MFG: Lattice Semiconductor
1	Raspberry Pi CM5	PN: SC1601, MFG: Raspberry Pi
1	Raspberry Pi CM5 Passive Cooler	PN: SC1752, MFG: Raspberry Pi
1	IMX219 Camera with regular (narrow) FOV	PN: (ArduCam B0191), MFG: Arducam
1	22-22 RPi 5 Camera Cable 200 mm	PN: B0BQ72Q33C, MFG: SHIVLAHERI
1	USB Cable Type A Male to Type C Male 3.00'	PN: U428-003-G2, MFG: Eaton Tripp Lite
1	IMX219 Camera holder	PN: R213343 (web store SKU), MFG: KKSB
1	5 V 25 W AC/DC External Wall Mount (Class II) Adapter Multi-Blade (Sold Separately) Input	PN: WR9QA5000USBCNAN(R6B), MFG: GlobTek, Inc.
1	Input Plug for Wall Adapter	PN: Q-KIT®, MFG: GlobTek, Inc.
1	Modular Cable Plug to Plug 8p8c (RJ45, Ethernet) 3.28' (1.00 m) Shielded	PN: BC-5SG010M, MFG: Bel Inc.
1	USB 2.0 Cable Type A Male to Micro B Male 3.28' (1.00 m) Shielded	PN: CBL-UA-MUB-1 MFG: Harwin Inc.
3	Conn Shunt 1.27 mm black handle – Used to enable eMMC write mode on the CM5	PN: M50-2000005 MFG: Harwin Inc.

5.1.2. Software

The pre-installed CrossLinkU-NX SoM HMI demo package includes:

- Raspberry Pi OS – Bookworm
- Lattice EVE SDK
- Lattice Web Server

Below is the demo bring-up utility:

- PuTTY

The following are the system recovery tools:

- Lattice Radiant Programmer 2025.1 and above
- Raspberry Pi Imager v1.9.6 and above
- Raspberry rpiboot

5.2. Board Setup

The following sections show the available boards.

5.2.1. CrossLinkU-NX SoM HW Board

The CrossLinkU-NX SoM HW board (LIFCL-33U-SOM-EVN) is a compact SoM HW board that handles the MIPI CSI-2 camera input, AI inference pipeline, and low-power management. It includes connectors for Camera Sensor (*J6*), USB-C (*J3*), header pins for JTAG programming (*J7*-Reserved) and board to board connector to generic carrier HW board (*J1* and *J2*).

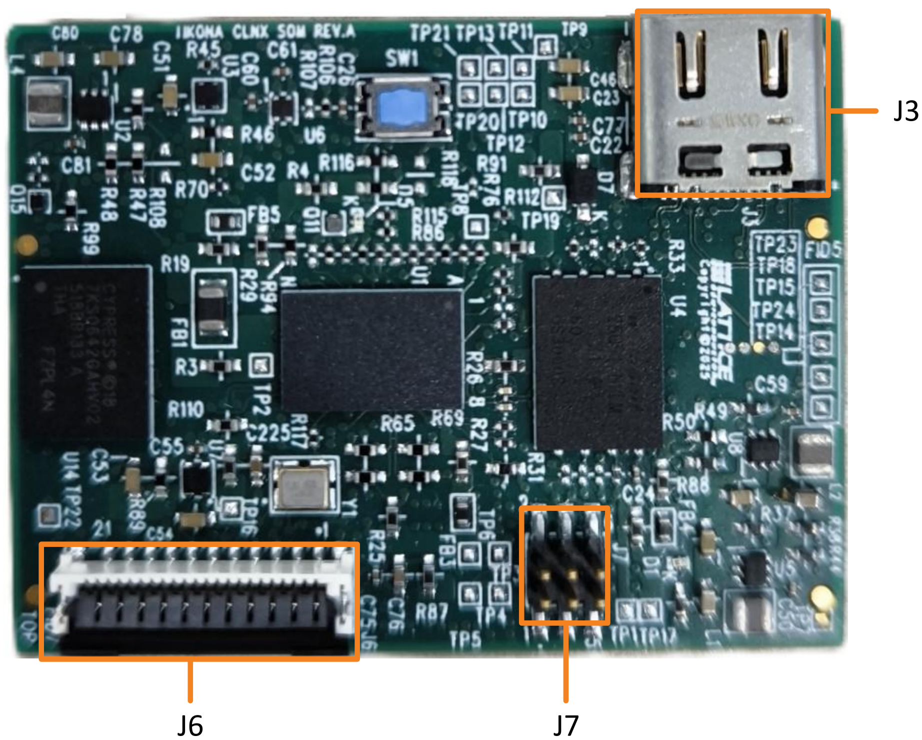

Figure 5.1. CrossLinkU-NX SoM HW Board (Top View)

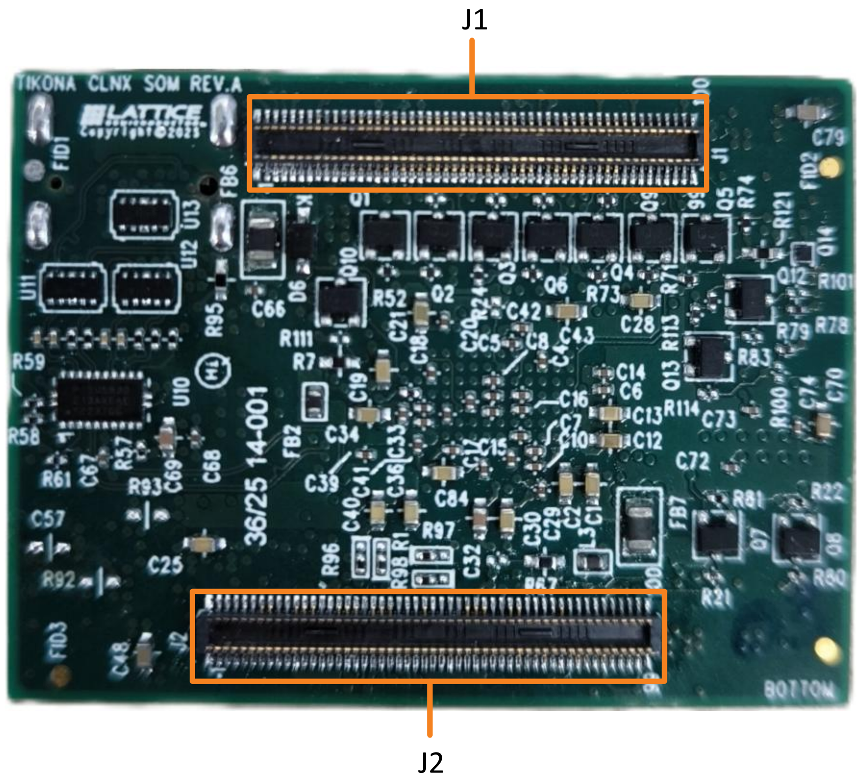

Figure 5.2. CrossLinkU-NX SoM HW Board (Bottom View)

5.2.2. Generic Carrier HW Board

The Generic Carrier HW board (LF-GEN-CR-PCBA) acts as a central interface, bridging the CrossLinkU-NX SoM and CM5. The key connections include:

- Power Connector, USB and Ethernet Ports: The carrier board routes the USB Type C (*J19*) for power and host communication, USB Type A (*J13*) for firmware update and RJ45 Ethernet connections (*J1*) for network access.
- UART and JTAG Interfaces: FTDI controller on the carrier board provide UART console access for CM5 and JTAG configurations for CrossLinkU-NX SoM. It is accessed through USB Micro Type B port (*J16*).
- Board to board connectors: One pair for interfacing with CrossLinkU-NX SoM board (*J14* and *J15*). The second pair for interfacing Raspberry Pi CM5 (*J6* and *J7*).

WiFi option is disabled by default.

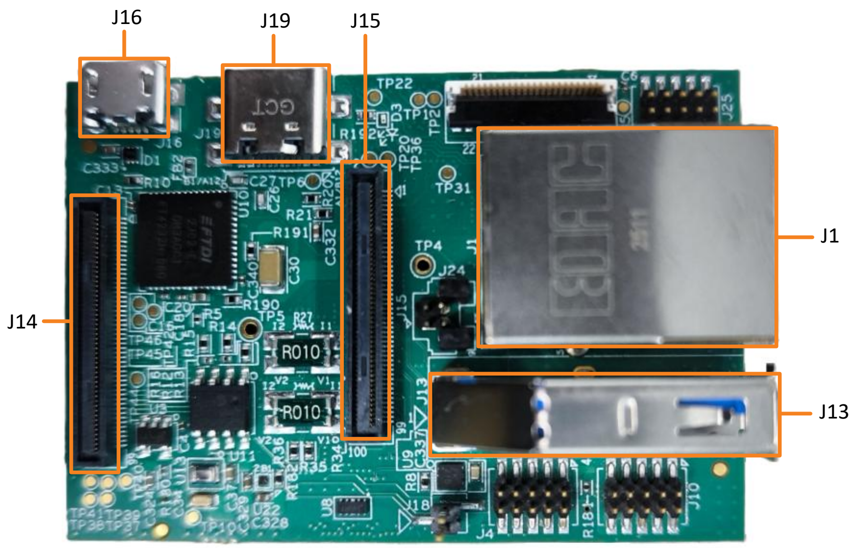

Figure 5.3. Generic Carrier HW Board (Top View)

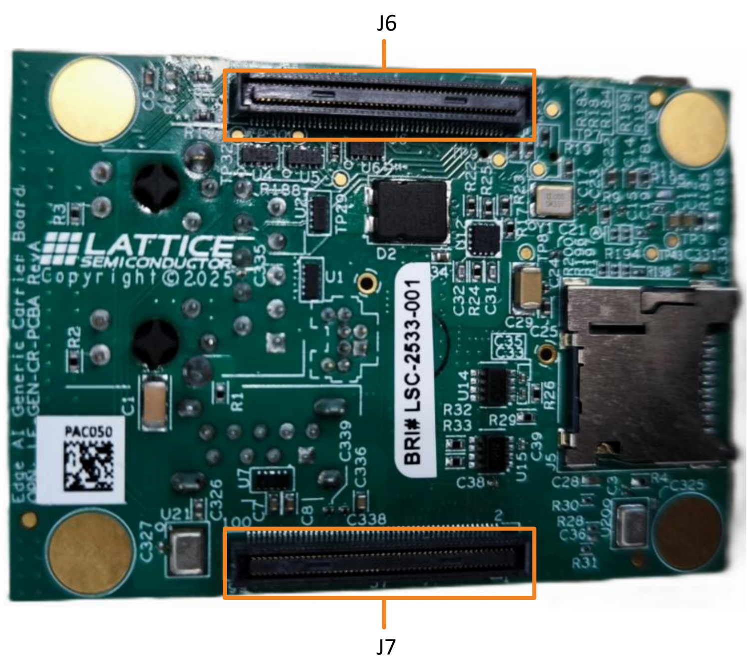

Figure 5.4. Generic Carrier HW Board (Bottom View)

5.2.3. Raspberry Pi CM5

The host compute module running the Raspberry Pi OS handles the AI metadata processing with EVE SDK and provides visualization dashboard through web server. The host module is equipped with USB, Ethernet, and GPIO interfaces.

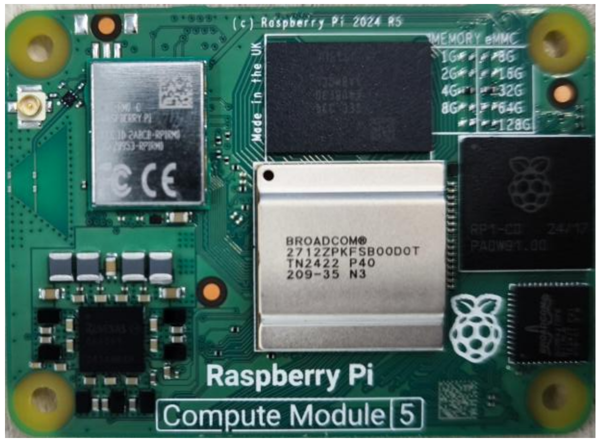

Figure 5.5. Raspberry Pi CM5 (Top View)

Figure 5.6. Raspberry Pi CM5 (Bottom View)

5.2.4. IMX219 Camera Module (Narrow FOV – ArduCam B0191)

The IMX219 Camera Module is a MIPI CSI-2 camera sensor that supports RAW10 format. This is used for video streaming and AI inferencing.

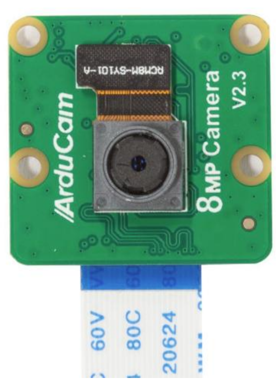

Figure 5.7. IMX219 Camera Module (Top View)

Figure 5.8. IMX219 Camera Module (Bottom View)

5.3. Board-to-Board Connections

This section describes the physical interconnections between the CrossLinkU-NX SoM HW board, the generic carrier HW board, Raspberry Pi CM5, and the camera module. Proper board-to-board connectivity is essential to ensure reliable data transfer for MIPI streaming and AI inferencing.

Figure 5.9 shows the physical stacking and orientation of the three main hardware boards used in the CrossLinkU-NX SoM HMI demonstration design.

To connect the board, perform the following:

1. From the right side of Figure 5.9, mount the **CrossLinkU-NX SoM HW** board onto the **Generic Carrier HW** board using the designated board-to-board connectors**.**
2. Mount the **Generic Carrier HW** board (with the **CrossLinkU-NX SoM HW** board attached) onto the **RPi CM5**.

All boards must be stacked vertically without changing orientation as depicted in Figure 5.9, ensuring that all board-to-board connectors are fully engaged to provide secure electrical and mechanical integration.

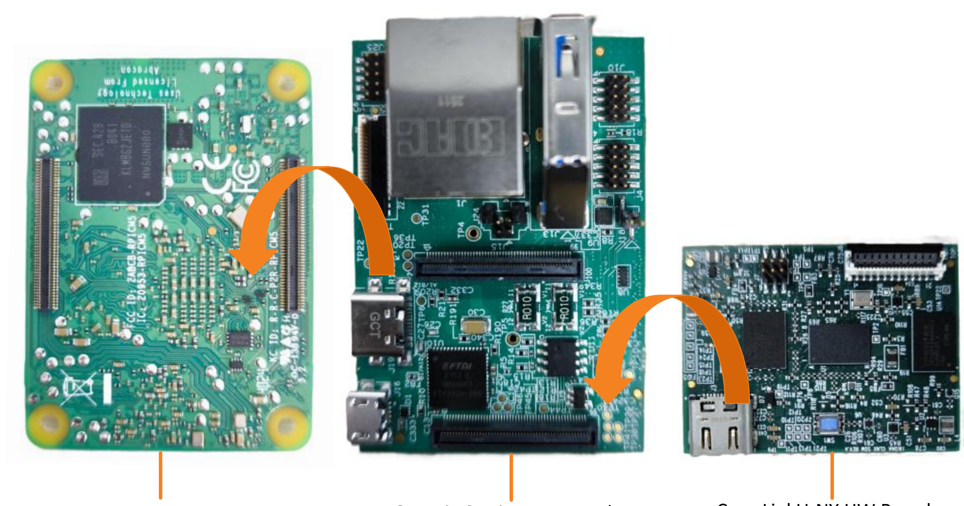

Figure 5.9. Stacking Orientation Arrangement

Figure 5.10 shows the fully assembled stack of hardware boards. Before proceeding, verify that all boards are correctly aligned and firmly seated in their respective connectors to ensure secure electrical and mechanical integration.

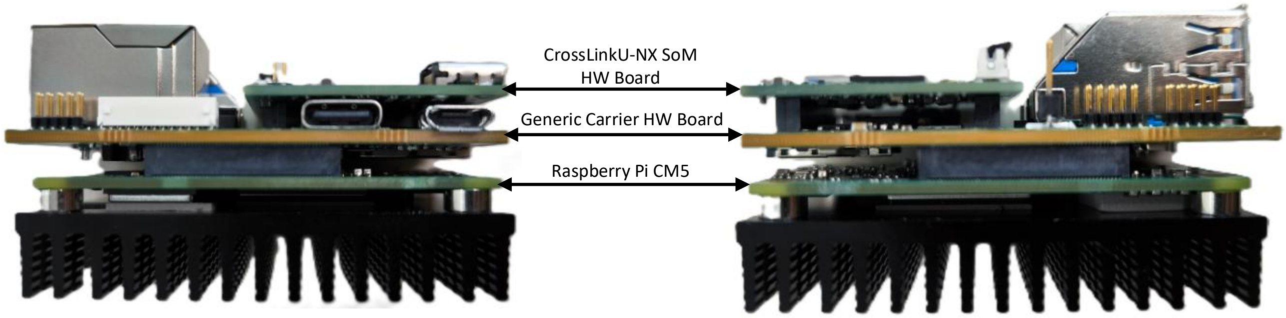

Figure 5.10. Board-to-Board Connections

The camera module connects to the CrossLinkU-NX SoM HW board through an FFC cable. Ensure the cable's contact is correctly aligned with the MIPI CSI-2 connector. The blue stiffener side must face the connector latch, as shown in Figure 5.11. To secure the cable, lift the connector latch, insert the cable fully and press the latch down to lock it in place. Avoid sharp bends in the FFC cable as it is fragile and may be damaged by excessive stress. Before powering the system, verify all connections to prevent damage or communication errors.

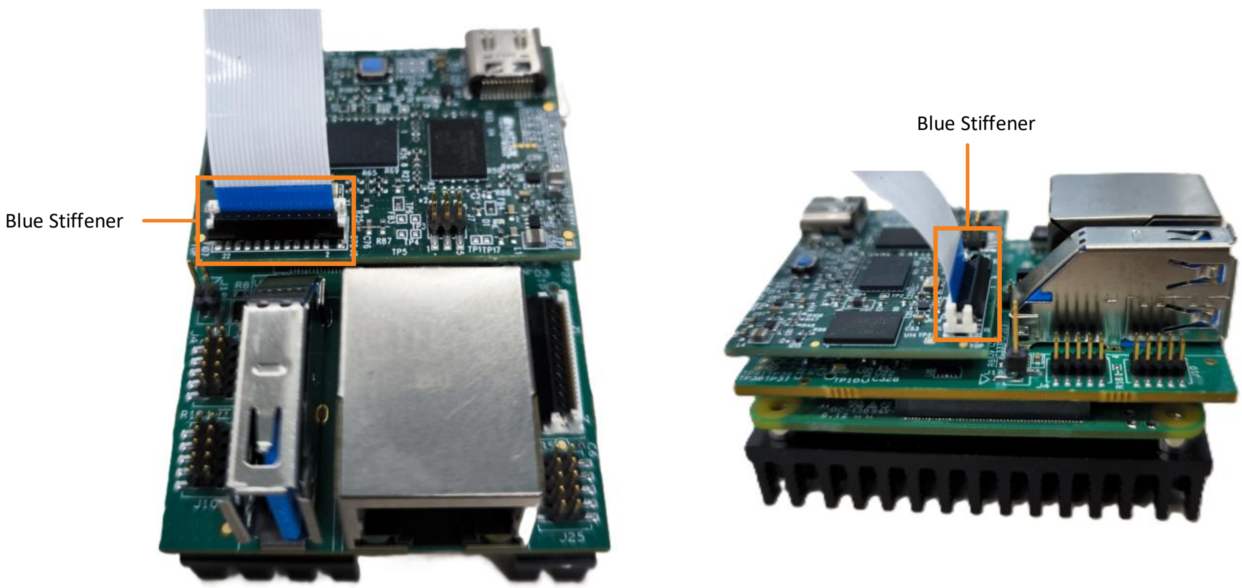

Figure 5.11. Board Connections with Camera FFC Cable

5.4. Configuring the Hardware Jumper

To configure the hardware jumper, connect the pin header *3 and 7* and *6 and 8* on the *J24* jumper as Figure 5.12. This allows the RPi to receive UART from the CrossLinkU-NX SoM.

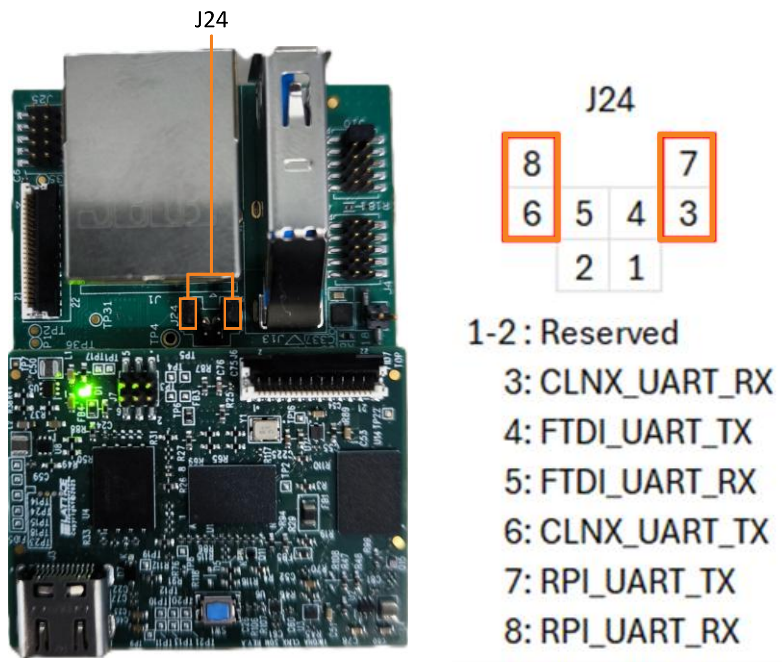

Figure 5.12. Hardware Connections for UART

5.5. RPi CM5 Serial Console Access

This section explains how to access Raspberry Pi CM5's Linux terminal through its built-in serial console. This allows for network-free headless access. The minimal connection required for serial console access is shown in Figure 5.13. A USB Type C power brick provides power to the module while a USB Type A to Micro Type B cable provides connectivity from the Host PC to Raspberry Pi CM5.

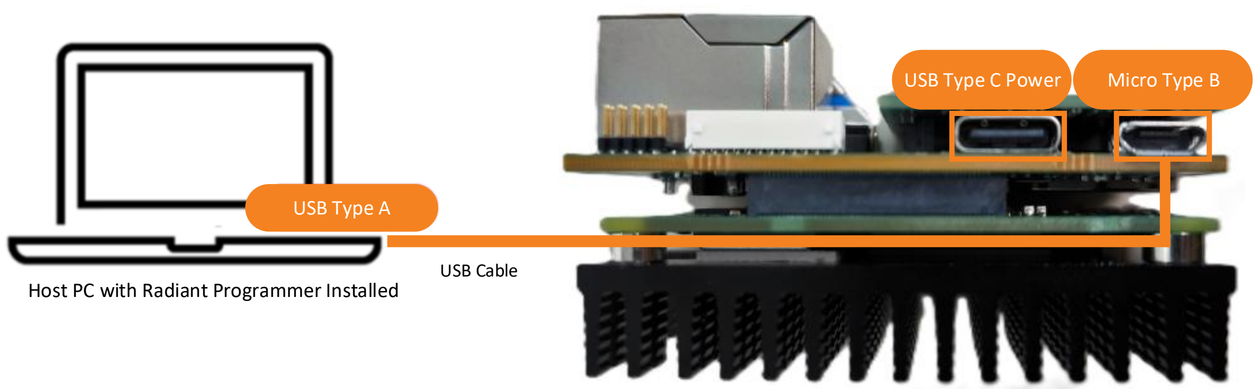

Figure 5.13. Hardware Setup for Serial Console Access

To access the Raspberry Pi console terminal, perform the following:

1. Connect the board to the PC using a USB Type A to Micro Type B cable, as shown in Figure 5.13. Connect the power adapter to the Raspberry Pi CM5 board and power on the adapter.
2. After powering on the module, the required COM port for serial console access is shown in the host PC's device manager. There are four new COM ports available to select. Open each of the *Properties* window to get the location name of **on USB Serial Converter D**.

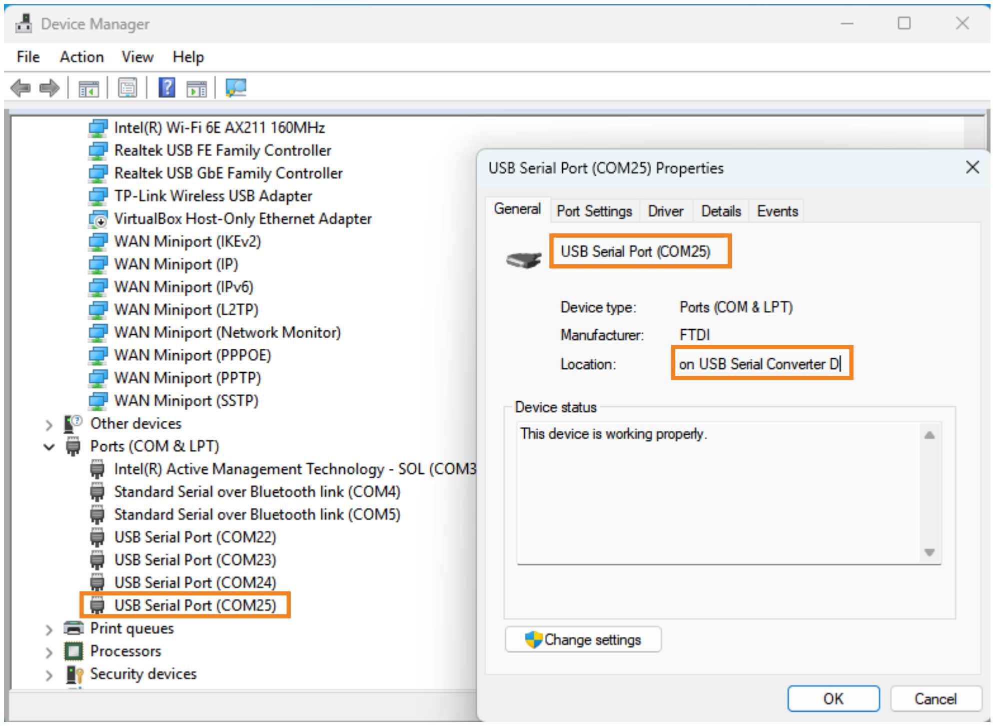

Figure 5.14. Available COM Ports in Device Manager

3. Follow the steps as shown in Figure 5.15 to open the selected COM port with PuTTY. COM settings for PuTTY are stated below.

```
Baud Rate:      115200
Data Bits:      8
Parity:         None
Stop Bits:      1
Flow Control:   None
```

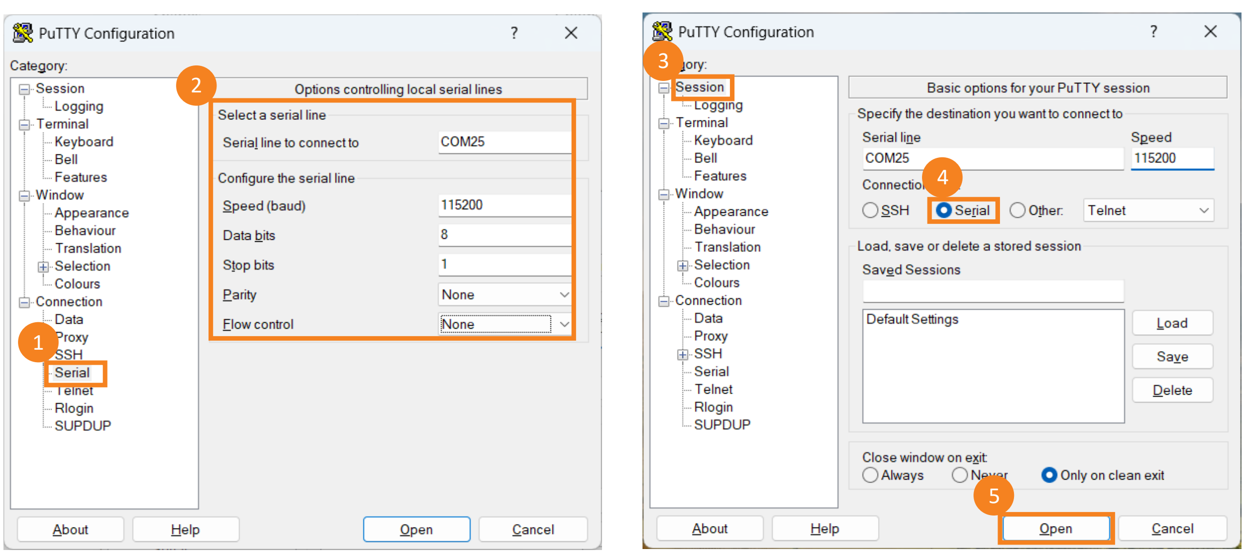

Figure 5.15. PuTTY Configuration to Open COM Port

4. After opening serial port connection, there must be a blank screen only. Press **Enter** to send a character and the login prompt is displayed as shown in Figure 5.16.

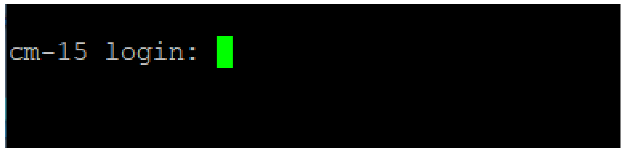

Figure 5.16. RPi CM5 Login Screen

5. Log in with the default credentials. A terminal prompt is displayed as shown in Figure 5.17.

```
Username: lattice
Password: lattice
```

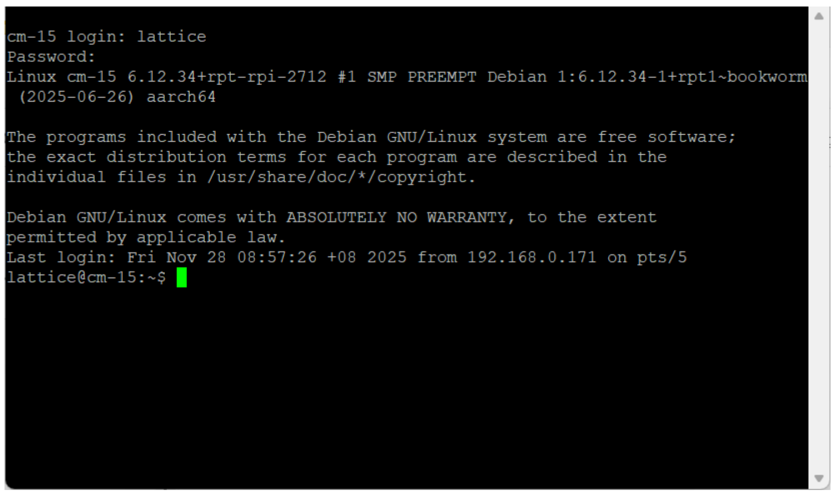

Figure 5.17. Serial Terminal Prompt

5.6. Connecting the RPi CM5's Ethernet

This section explains how to obtain the IP address of RPi CM5 for remote access. The Raspberry Pi CM5's ethernet can be connected to the host PC using either of the following ways:

- Using a local network setup on a router (refer to Setting Up Local Network on Router section) or,
- Using a direct connection (refer to Setting Up Direct Connection to Host PC section)
  - direct connection with Static IP or,
  - direct connection with Dynamic IP

Note: Wi-Fi option is disabled by default on generic carrier board.

To set up the pre-requisite for the Ethernet connection, perform the following:

1. Connect the CrossLinkU-NX SoM kit's LAN connection (as shown in Figure 5.18) to either the host PC or a router with DHCP enabled.

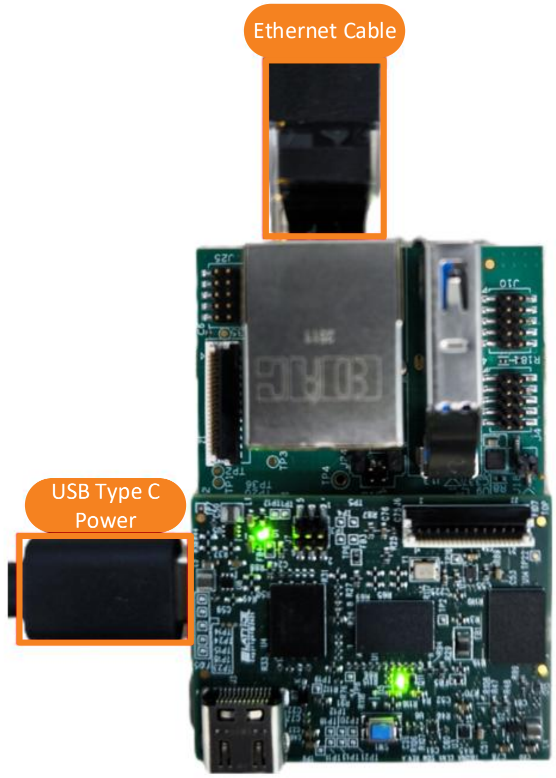

Figure 5.18. CrossLinkU-NX SoM with Ethernet Connection

2. Navigate to RPi Serial Console and running the subsequent commands.
   - Run this command:

```
sudo nmcli connection show
```

   - The output has a format like the example below. Find the Ethernet connection name and device (*Wired connection 1* and *eth0* in the example below.

```
NAME                UUID                                  TYPE      DEVICE
lo                  a7229c70-ee32-4030-b548-cb69435559bc  loopback  lo
Wired connection 1  07561947-8e94-4a72-b0d5-1317f8821845  ethernet  eth0
```

   - Define these environment variables:

```
export CONNECTION_NAME={Connection name like "Wired connection 1"}
export DEVICE={Device name like "eth0"}
```

   For example:

```
export CONNECTION_NAME="Wired connection 1"
export DEVICE="eth0"
```

If you connected your Ethernet cable to a router, skip to section Setting Up Local Network on Router. If you connected to the Host PC, refer to the Setting Up Direct Connection to Host PC section.

If no internet connection is needed on the RPi CM5, you may refer to the Setting Up Direct Connection to Host PC section with Direct Connection with Static IP for the most reliable method.

5.6.1. Setting Up Local Network on Router

The pre-requisite for this setup is a router with DHCP enabled.

To set up the local network connection on the router, perform the following:

1. Connect the host PC to the router through ethernet or Wi-Fi connection.
2. Run these commands on PuTTY terminal to make sure that the connection IP is dynamic:

```
sudo nmcli connection delete "$CONNECTION_NAME"
sudo nmcli connection add type ethernet ifname $DEVICE con-name "$CONNECTION_NAME"
ipv4.method auto
```

3. Unplug the ethernet cable and plug it again.

5.6.2. Setting Up Direct Connection to Host PC

There are two ways to set up direct LAN connection to Host PC.

- Direct Connection with Static IP or,
  - Pre-requisite is that the host PC must have one LAN adapter.
- Direct Connection with Dynamic IP
  - The pre-requisite for this setup is that the host PC must have two network adapters, using either following combination:
    - Two LAN adapters (one for connecting to RPi CM5 and the other one for connecting to Internet) or,
    - One LAN adapter and one Wi-Fi adapter (LAN adapter for connecting to RPi CM5 and Wi-Fi adapter for connecting to Internet)

If you need internet on the RPi CM5, you need to follow the Dynamic IP method.

5.6.2.1. Direct Connection with Static IP

Follow these steps only if the direct connection with static IP is selected:

1. Open the **Control Panel** on the host PC.
2. Click **Network and Internet** as shown in Figure 5.19.

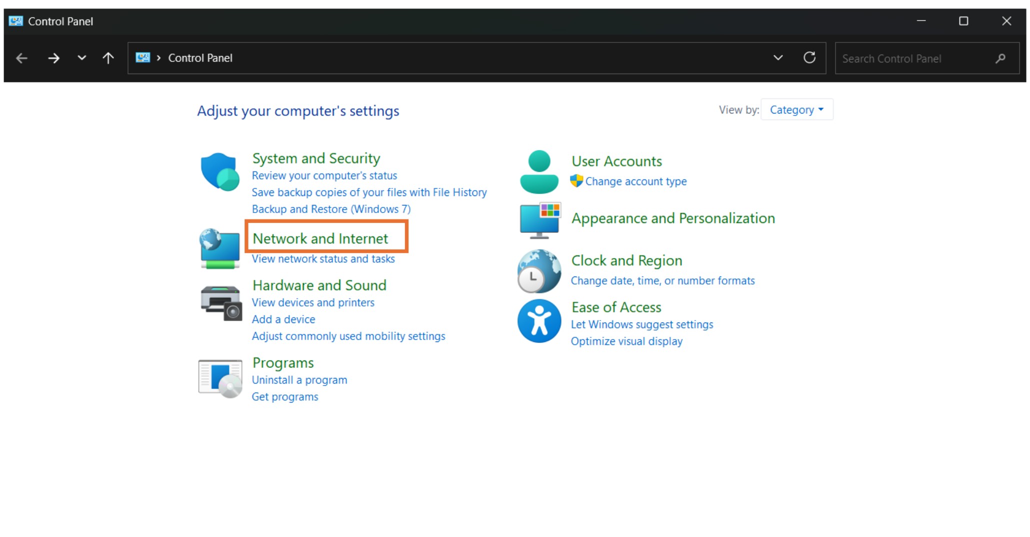

Figure 5.19. Control Panel

3. Click **Network and Sharing Center** as shown in Figure 5.20.

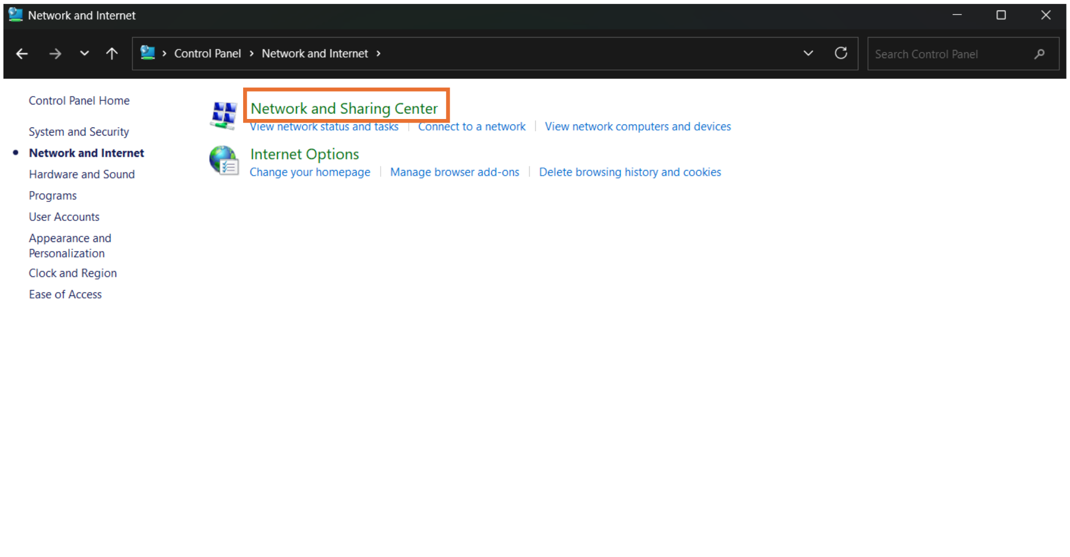

Figure 5.20. Network and Internet

4. Click **Change adapter settings** on the right side as shown in Figure 5.21.

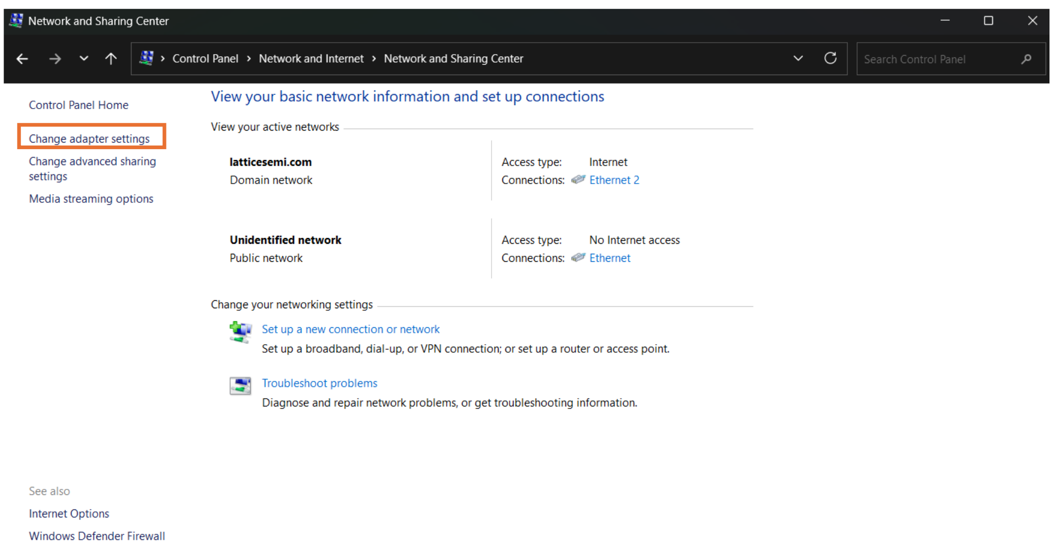

Figure 5.21. Network and Sharing Center Page

5. Right-click on the adapter which is connected to the internet and select **Properties** as shown in Figure 5.22.

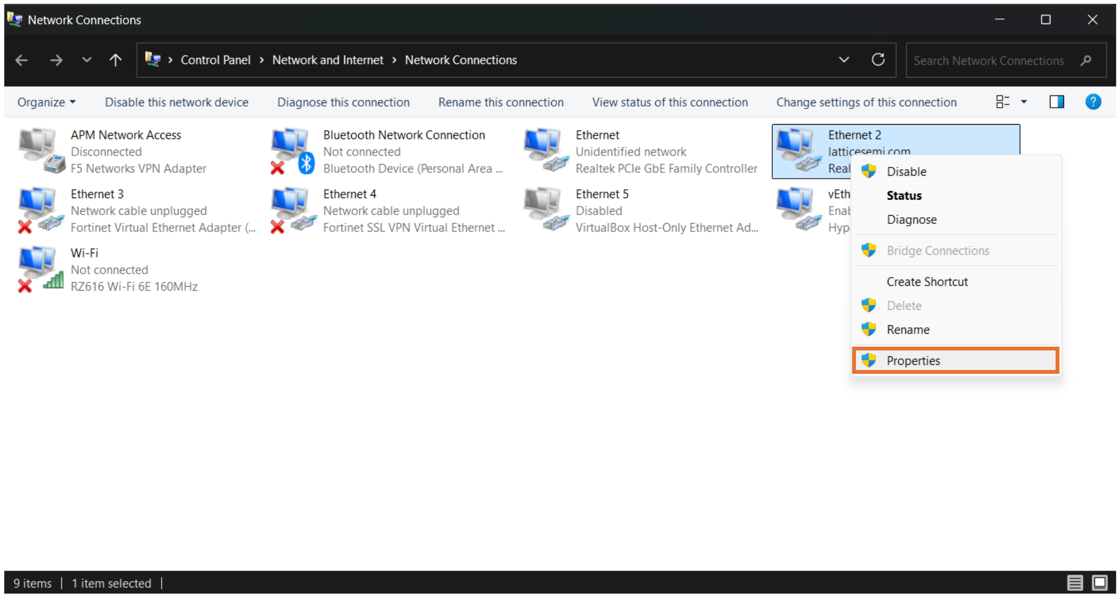

Figure 5.22. Network Connections

6. Select Internet Protocol Version 4(TCP/IPv4). Then click on the Properties button following Figure 5.23.

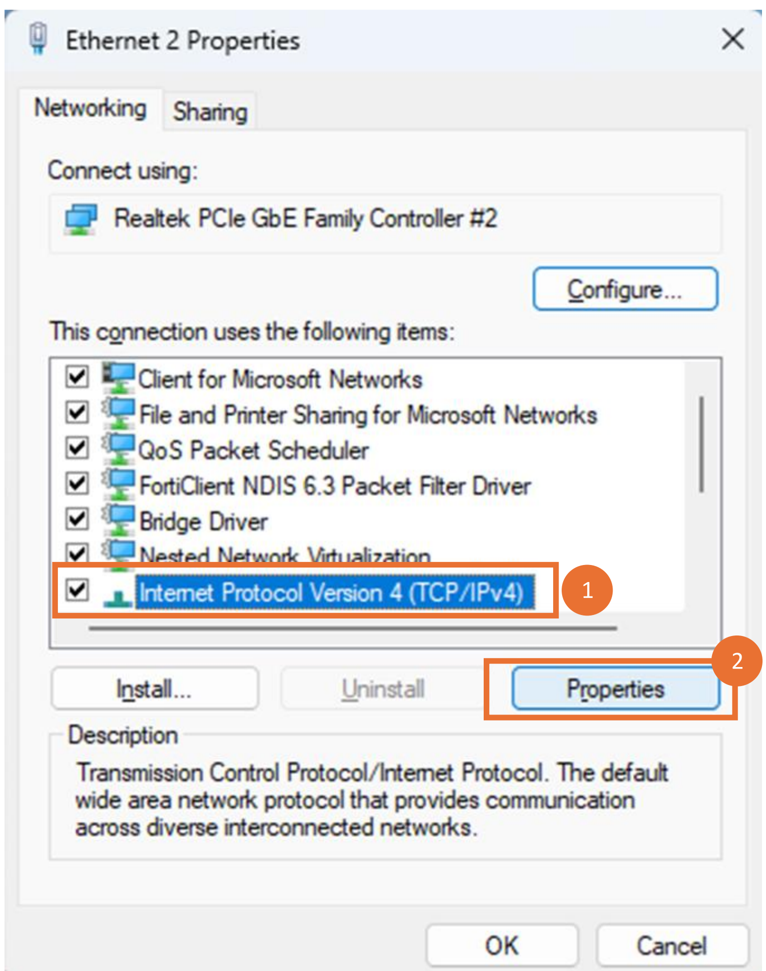

Figure 5.23. Changing Internet Protocol Version 4(TCP/IPv4) Settings

7. Fill the properties as shown in Figure 5.24.

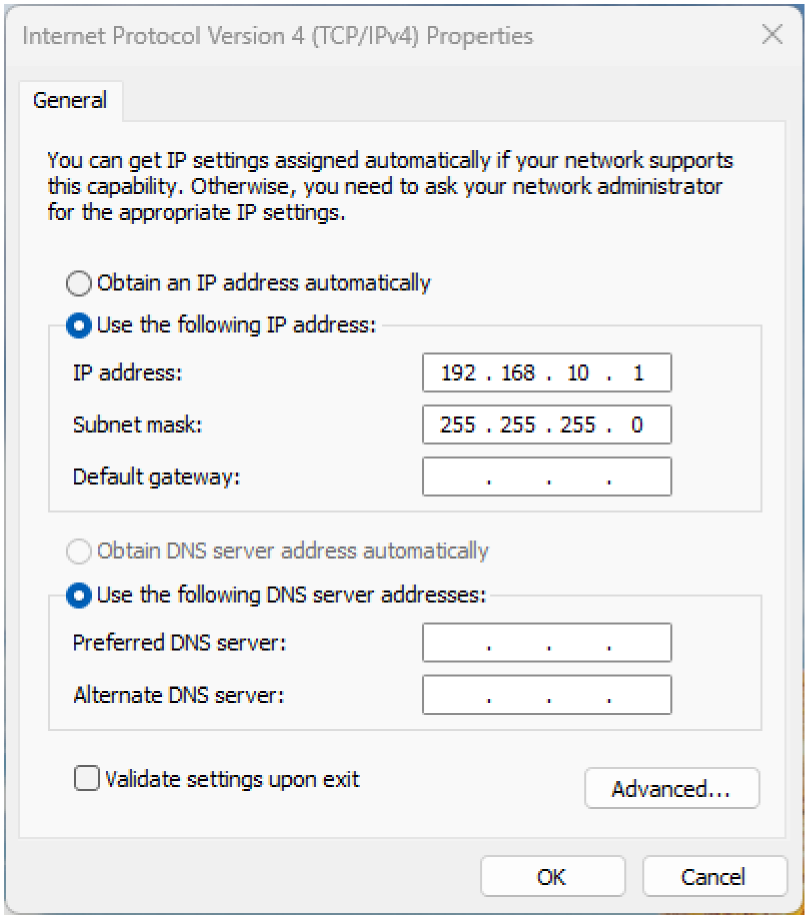

Figure 5.24. Static IP Settings

8. Set the static IP on the Raspberry Pi using following command on the PuTTY terminal:

```
sudo nmcli connection modify "$CONNECTION_NAME" \
  ipv4.method manual \
  ipv4.addresses "192.168.10.2/24" \
  ipv4.gateway "" \
   ipv4.dns ""
```

9. Unplug the ethernet cable and plug it again.
10. Disable internet on PC to avoid routing confusion.
11. Ping the PC from Raspberry Pi using this command on PuTTY terminal to make sure that the data is transferred successfully:

```
ping -c 3 192.168.10.1
```

12. Ping the Raspberry Pi from PC using this command on PuTTY terminal to make sure that the data is transferred successfully:

```
ping 192.168.10.2
```

5.6.2.2. Direct Connection with Dynamic IP

The following steps must be followed only if the PC is connected to the local network and the dynamic IP method is chosen to be used.

1. Open the **Control Panel** on the host PC.
2. Click **Network and Internet** as shown in Figure 5.19.

3. Click **Network and Sharing Center** as shown in Figure 5.20.
4. Click **Change adapter settings** on the right side as shown in Figure 5.21.
5. Right-click on the adapter which is connected to the internet and select **Properties** as shown in Figure 5.22.
6. Share the connection as shown in Figure 5.25:
   a. Select **Sharing** tab.
   b. Check the Allow other network users to connect through this computer's Internet connection option.
   c. Select the adapter which is connected to the Raspberry Pi as **Home networking connection**. The drop-down menu may not appear if there are only two network adapters.
   d. Uncheck **Allow other network users to control or disable the shared Internet connection** option.
   e. Select **OK**.

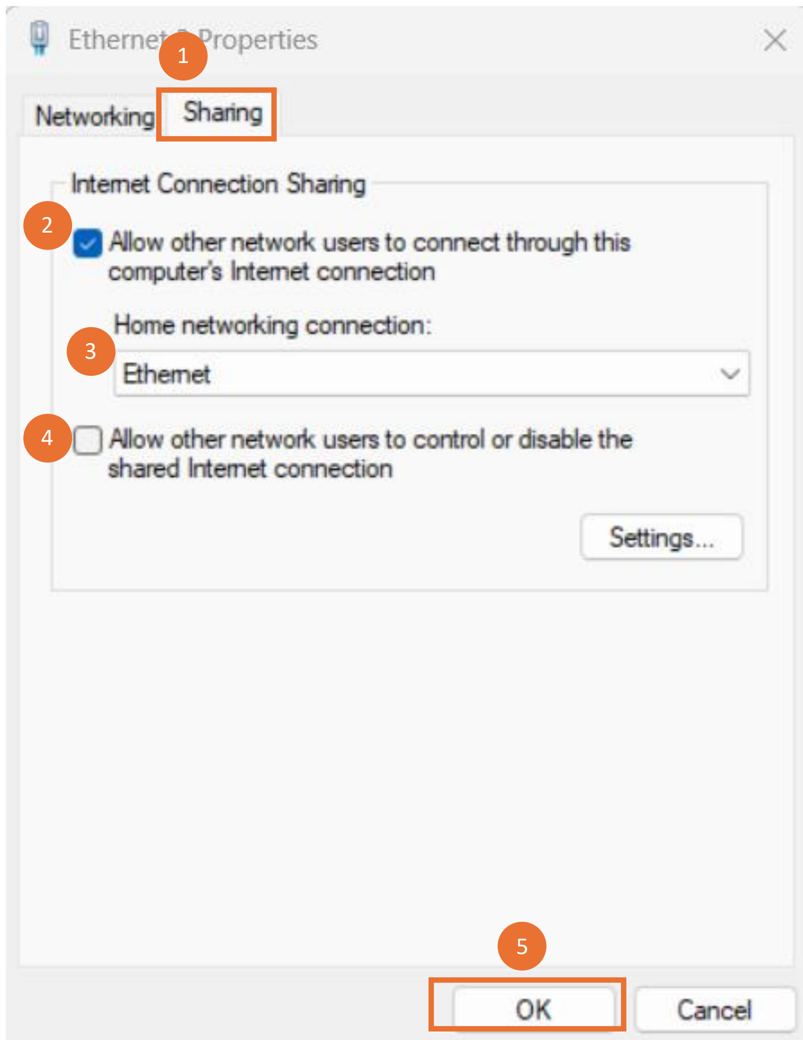

Figure 5.25. Network Adapter Properties

7. Run these commands on PuTTY terminal to make sure that the connection IP is dynamic:

```
sudo nmcli connection delete "$CONNECTION_NAME"
sudo nmcli connection add type ethernet ifname $DEVICE con-name "$CONNECTION_NAME"
  ipv4.method auto
```

8. Unplug the ethernet cable and plug it again.

5.6.3. Obtaining IP Address of Raspberry Pi CM5

After completing the steps in Setting Up Local Network on Router or Setting Up Direct Connection to Host PC, run command below in the terminal to obtain the *IP address* of the RPi CM5 as shown in Figure 5.30.

```
ifconfig
```

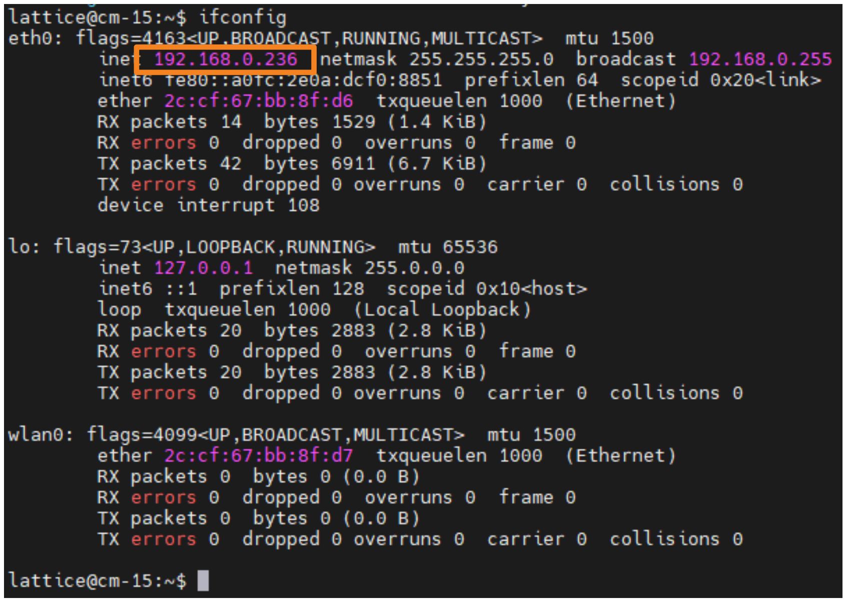

Figure 5.30. Obtained IP Address from ifconfig Command

Note: For terminal access, refer to the RPi CM5 Serial Console Access section.

5.7. Running EVE and Web Server

To run the EVE and web server, perform the following:

1. For terminal access, refer to the RPi CM5 Serial Console Access section.
2. Start the web server.

   ```
   run_eve_web_server
   ```

   Note the **IP address** or the **URL displayed** in the logs as shown in Figure 5.31.

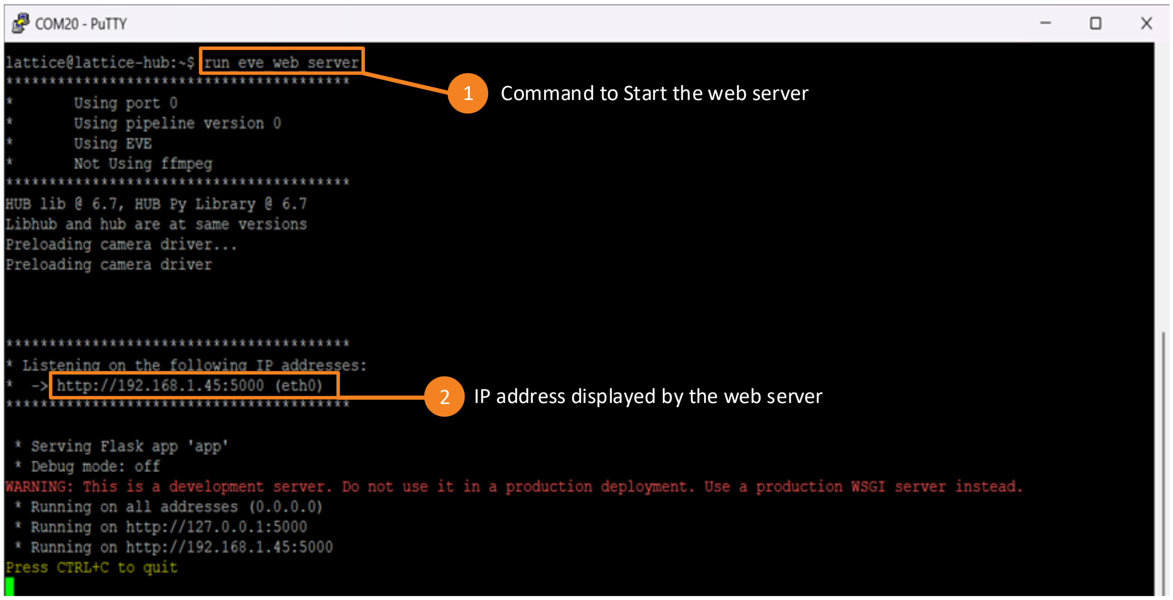

Figure 5.31. IP Address of Web Server

3. In your host computer, open a web browser and enter the address obtained in Step 2. For details on navigating the Web App interface, refer to the Web Server section.

Notes:

1. Terminate the Web Server: After completing the operation, terminate the web server by pressing CTRL+C in the terminal where the application was launched.
2. Switching Modes: To switch between streaming and sensing modes, first terminate the web server. Then, relaunch the web server to return to the landing page.

6. Running the Firmware Management Application

Before executing the application, make sure the web server is not running.

6.1. Updating the Firmware

To update the firmware on the FPGA, perform the following steps:

1. Connect the USB Type C to Type A cable from USB Type C connector (J3) on the CrossLinkU-NX SoM HW board to USB Type A connector (J13) on the carrier board. This connects RPi CM5's USB to CrossLinkU-NX SoM as shown in Figure 6.1.

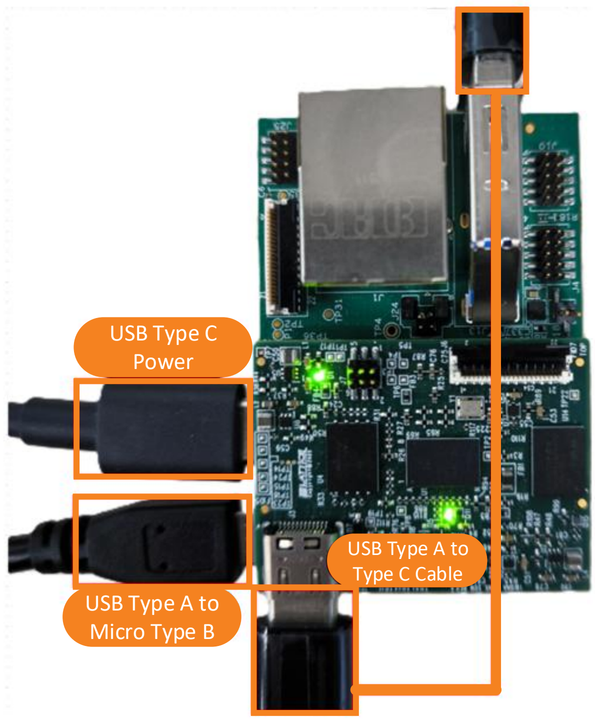

Figure 6.1. Hardware Connections for Firmware Update

2. For terminal access, refer to the RPi CM5 Serial Console Access section.
3. Navigate to the **FW_UPDATE_TOOL** folder (Path: */opt/clnx_som/FW_Update*) to access the Lattice Firmware Management Application.
4. Run the executable.
```
sudo ./clnxsomdemo_fwupd_<firmware_version>
Example: sudo ./clnxsomdemo_fwupd_01.12.00.00
```
5. Wait for the update to be completed. The executables display the progress in the terminal. Do not disconnect the device or power off during this step.
6. Once the update is completed, it displays the success message shown in Figure 6.2. Power cycle the board after programming.

```
2025-12-04 14:55:33,524 [INFO] [FwUpd_01.02.00.00] Starting SoM Firmware Update to 01.12.00.00
2025-12-04 14:55:34,106 [INFO] Updating Current NX        [5403] to Target [5405]
2025-12-04 14:55:52,251 [INFO] Skipping update for FM        (Version: 5303)
2025-12-04 14:55:52,251 [INFO] Updating Current RISCV     [5506] to Target [550F]
2025-12-04 14:55:57,421 [INFO] Skipping update for FD        (Version: 0115)
2025-12-04 14:55:57,422 [INFO] Skipping update for FID       (Version: 0511)
2025-12-04 14:55:57,423 [INFO] Skipping update for LV        (Version: 0412)
2025-12-04 14:55:57,423 [INFO] Skipping update for PARSER    (Version: 0903)
2025-12-04 14:55:57,424 [INFO] Updating Current PD        [0612] to Target [0614]
2025-12-04 14:56:13,010 [INFO] Update completed successfully.
```

Figure 6.2. Firmware Update Success Message

6.2. Firmware Version

The following steps queries the bitstream version currently on the hardware device.

1. Follow steps 1 to 3 in the Updating the Firmware section.
2. Run the executable.

```
sudo ./clnxsomdemo_fwupd_<firmware_version> -v
Example: sudo ./clnxsomdemo_fwupd_01.12.00.00 -v
```

3. The executables display each of the bitstream versions in the terminal. Wait 30 seconds before running the web server.

```
2025-12-04 14:56:22,310 [INFO] ==================================================
2025-12-04 14:56:22,310 [INFO]   CLNX SoM Firmware Update Tool
2025-12-04 14:56:22,310 [INFO]   Version : 01.02.00.00
2025-12-04 14:56:22,310 [INFO] ==================================================
2025-12-04 14:56:22,310 [INFO] Global Version [01120000]
2025-12-04 14:56:22,310 [INFO] NX             [5405]
2025-12-04 14:56:22,310 [INFO] FM             [5303]
2025-12-04 14:56:22,310 [INFO] RISCV          [550F]
2025-12-04 14:56:22,311 [INFO] FD             [0115]
2025-12-04 14:56:22,311 [INFO] FID            [0511]
2025-12-04 14:56:22,311 [INFO] LV             [0412]
2025-12-04 14:56:22,311 [INFO] PARSER         [0903]
2025-12-04 14:56:22,311 [INFO] PD             [0614]
```

Figure 6.3. Firmware Versions

7. Debug Methodology

This section outlines the recommended steps for debugging the design. By following this section, system functionality can be verified and potential issues during integration and testing can be identified.

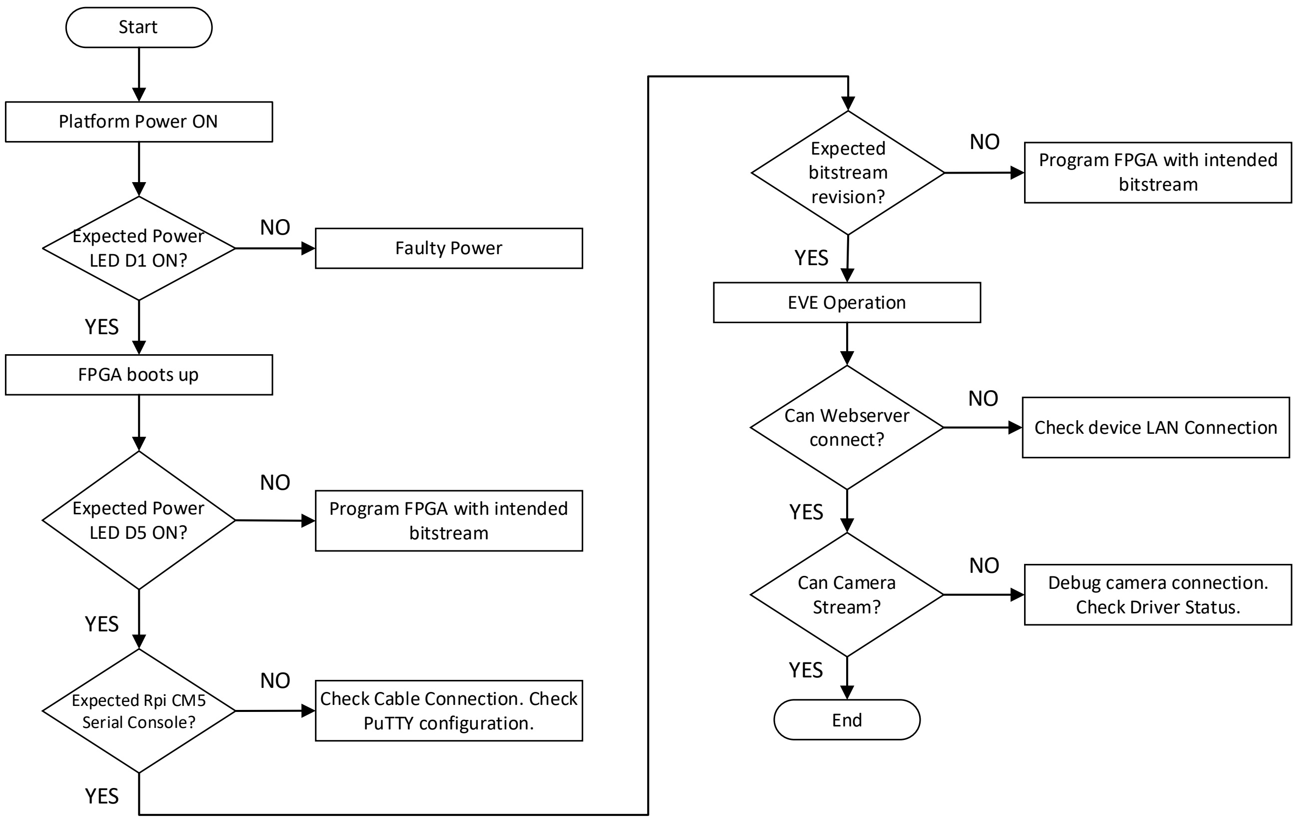

Figure 7.1. Debug Methodology Flow Chart

7.1. Powering Up the CrossLinkU-NX SoM

To power up the CrossLinkU-NX SoM, perform the following steps:

1. Power cycle the board.
2. This power cycle ensures the FPGA correctly loads and configures the new firmware.
3. Verify LEDs status.
4. Upon powering up, check the status of LEDs on the CrossLinkU-NX SoM HW board. Refer to Figure 7.2 for LED locations. These LEDs indicate the board's operation status. Table 7.1 describes each LED behavior upon boot up.

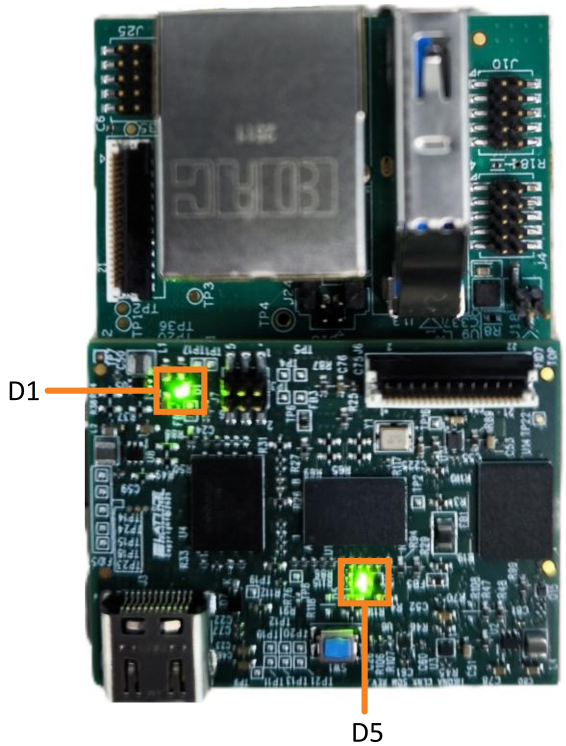

Figure 7.2. LED Position on CrossLinkU-NX SoM

Table 7.1. LED Status Upon Board Power Up

LED	Connected Signal	Board Power UP
D1	BOARD VCC Supply	ON
D5	CLNX_DONE	ON

7.1.1. Troubleshooting Common Issues

- Issue A: No LEDs are on.
  - Action 1: Check that the power supply for USB Type C cable is connected to the carrier board as shown in Figure 5.13 and turn on.
- Issue B: LED D5 is not on.
  - Action 1: Reprogram the FPGA per the Programming the Bit File section to confirm the bitstream is updated.
  - Action 2: If the issue persists, submit a technical support case through www.latticesemi.com/techsupport.

7.1.2. Programming the Bit File

This section explains how to program a bitstream file to the CrossLinkU-NX System-on-Module (SoM). In most cases, this step is not required, as the CrossLinkU-NX SoM is pre-programmed with the HMI demonstration bitstream from the factory. You only need to perform these steps if you need to reprogram the module due to updates or recovery.

To program the bit file, perform the following:

1. Connect the board to the PC using a USB Type A to Micro Type B cable, as shown in Figure 7.3. Connect the power adapter to the Raspberry Pi CM5 board and power on the adapter.

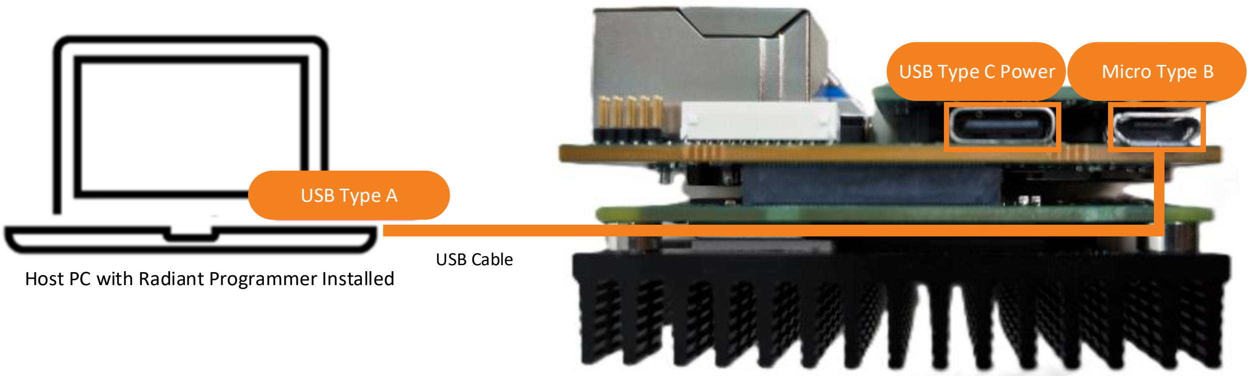

Figure 7.3. Hardware Setup for Bit File Programming

2. For terminal access, refer to the RPi CM5 Serial Console Access section.
3. Run the pre-programmer operation script.

```
pre_pgmr_op
```

4. Launch standalone Radiant Programmer on the Host PC. Enter the **Project Location** and **Project Name**; then click **OK**.

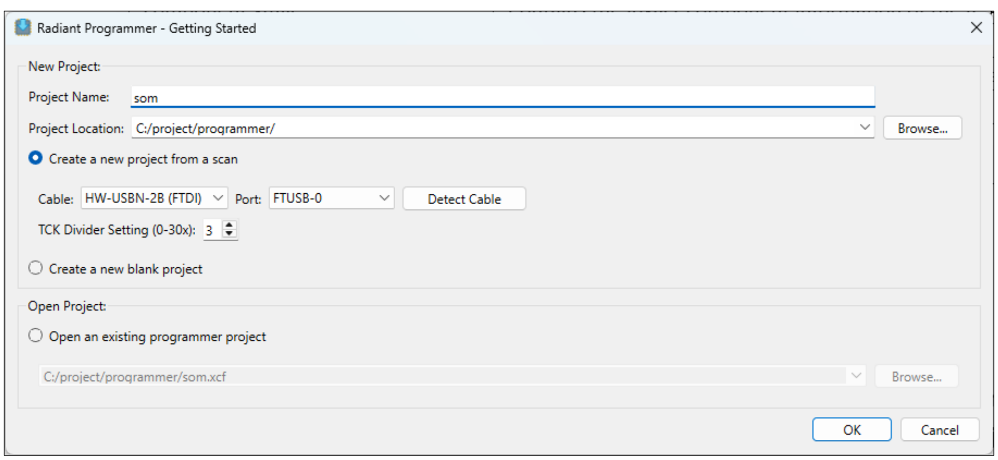

Figure 7.4. Radiant Programmer – Getting Started Window

5. Click **Detect Cable** in the cable setup section, select **FTUSB-0**, and then click **OK** to proceed.

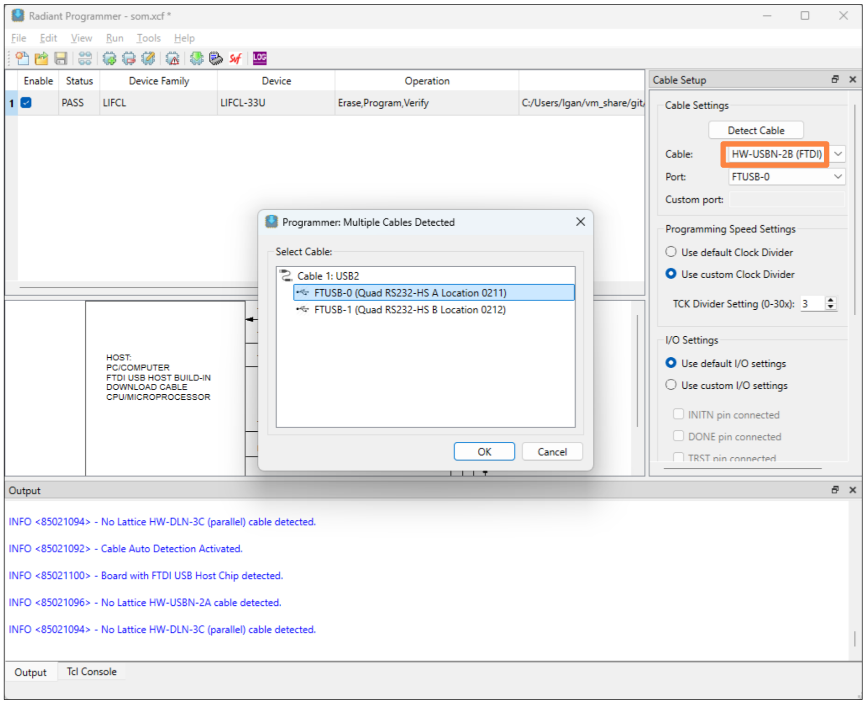

Figure 7.5. Radiant Programmer Detect Cable Settings

6. Double-click on **Erase, Program, Verify** in the **Operation** column to select the SPI Flash configuration and the bitstream file. The *Device Properties* window may take up to a minute to load.
7. Select the appropriate SPI flash configuration to flash directly to external SPI flash. Refer to Figure 7.6 for the expected settings. Click the **...** button beside the file name bar and select the bitstream file from the *CLNX_SPI_IMAGE* folder listed in Table 2.1.

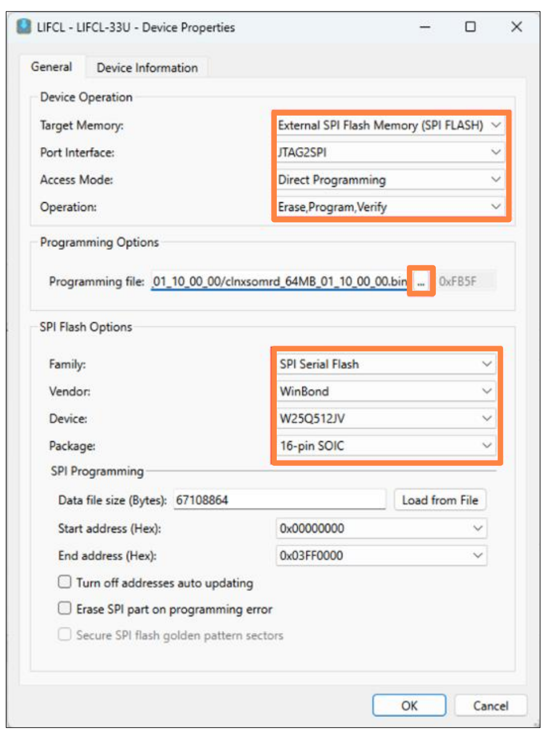

Figure 7.6 Selecting Bitstream File to Program

8. Click the **Program Device** toolbar icon to program the bitstream into the board, as shown in Figure 7.7. This process may take up to **20 minutes** to complete.

Figure 7.7 Program Device Toolbar Icon

9. Verify that the programming status displays *Operation successful* in the output window, as shown in Figure 7.8. To activate the flashed bitstream, power cycle the device.

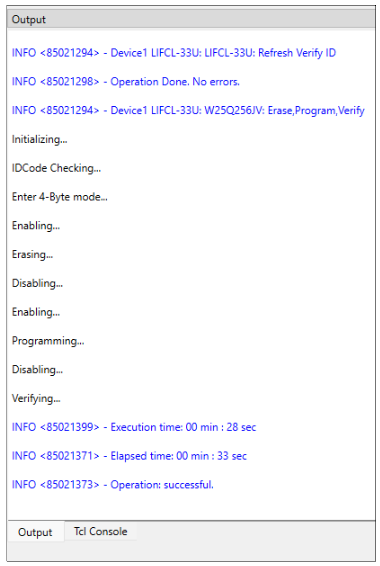

Figure 7.8. Message on Successful Programming

10. For terminal access, refer to the RPi CM5 Serial Console Access section.
11. Run the post-programmer operation script.

```
post_pgmr_op
```

7.2. Verifying Ethernet Connection

1. For terminal access, refer to RPi CM5 Serial Console Access.
   - On failure, check cable connection and PuTTY configuration.
   - If issue persists, refer to the Flashing the RPi CM5 section to recover the RPi CM5's OS.
2. To obtain the module's IP address, refer to Connecting the RPi CM5's Ethernet section for the steps.
3. Run command below on the host PC to check ethernet connectivity as shown in Figure 7.9.

```
ping <IP_Address>
```

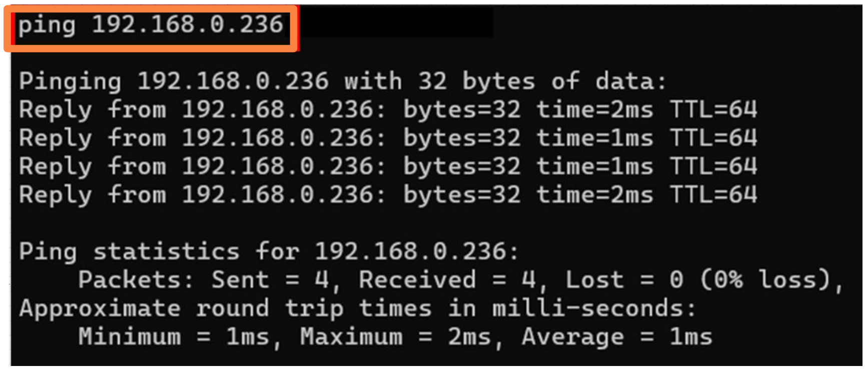

Figure 7.9. Verify Raspberry Pi CM5 Ethernet Connectivity

7.3. Flashing the RPi CM5

This section explains the steps required to flash the operating system onto the Raspberry Pi Compute Module 5 (CM5). In most cases, this step is not required, as the CM5 typically comes with a factory-installed OS image. You can skip this section unless you need to re-flash the module due to corruption or update requirements.

To flash the RPi CM5, perform the following:

1. Connect pin header **1 and 2** on **Jumper J10** as shown in Figure 7.10. This shorts the nRPIBOOT and enables flashing of the RPi CM5.

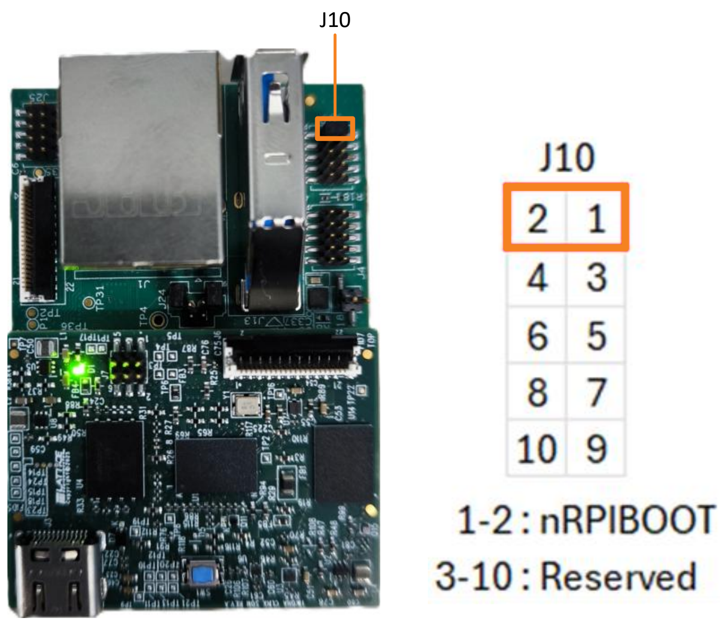

Figure 7.10. Hardware Connection for nRPIBOOT

2. Connect the board to the PC using a USB Type A to Type C cable as shown in Figure 7.11.

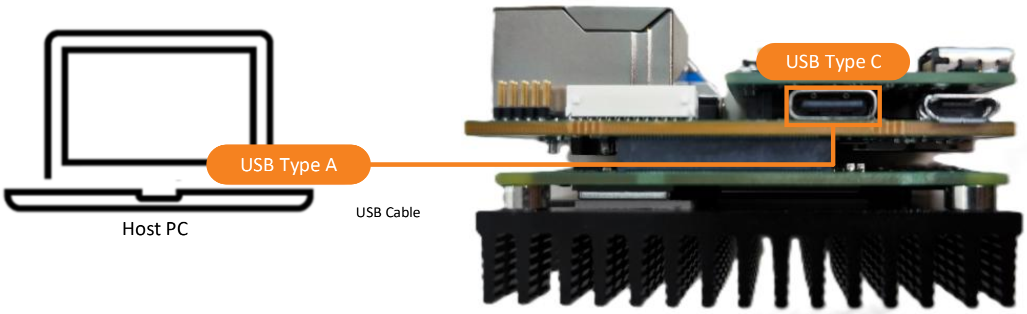

Figure 7.11. Hardware Setup for Flashing RPi CM5 OS

3. Run **rpiboot.exe** from the path *C:\Program Files (x86)\Raspberry Pi\rpiboot.exe*. This launches a Windows console, and if RPi CM5 is properly connected, the script continues to run and exits the terminal.

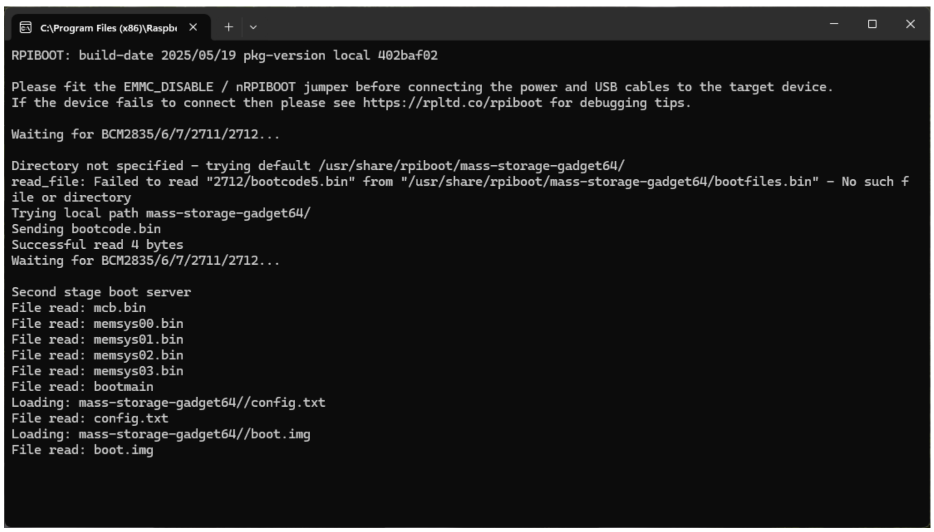

Figure 7.12. Configuring RPi CM5 as Mass Storage Device

4. The *Insert disk* window or drive content window may prompt when process is completed. Click **Cancel** or click *x* to close the prompt.

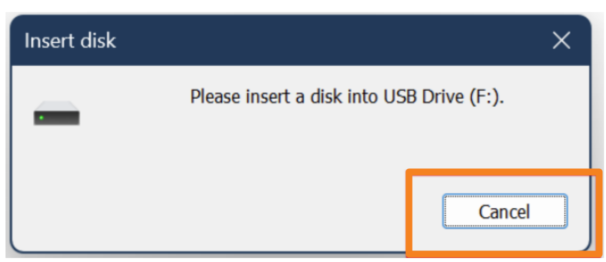

Figure 7.13. Pop Up to Insert Disk

5. Open Raspberry Pi Imager software. Ensure the **host PC** is connected to the **internet**.
6. Download the RPi OS image from RPI_OS folder listed in Table 2.1. Image format is *RPi_OS-arm64_<OS_Version>_CLNX_HMI_v< Edge_Hub_Version>.img.xz*. For example: **RPi_OS-arm64_1.0.0_CLNX_HMI_v5.img.xz**.
7. Click **Choose Device** and select **Raspberry Pi 5**.
8. Click **Choose OS**. Navigate to click the **Use custom** option and select the downloaded RPi OS image.

9. Click **Choose Storage** and select **mmcblk0 USB Device – 31.3 GB**. This must appear when rpiboot process is successful.
10. Click **Next**.
11. Select **No** on the **OS Customization** settings.
12. Start flashing by clicking **Yes**.
13. The flashing process takes approximately **15 minutes** to complete.
14. This writes the OS image to the RPi CM5, followed by verification.
15. Click **Continue** after the process completes.
16. Disconnect USB Type A to Type C cable from the PC and then only remove the jumper across pin header **1 and 2** on **Jumper J10.**

References

- CrossLinkU-NX HMI Demo web page
- CrossLinkU-NX SoM Board web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans
- Lattice Radiant Programmer Tools FPGA Programmer Standalone software for Windows
- Programming Tools User Guide for Radiant Software 2025.1
- Raspberry Pi Compute Module Hardware
- Raspberry Pi Imager v1.9.6 and above for Raspberry Pi's Imager tool
- Raspberry rpiboot for Raspberry Pi's rpiboot command-line utility

Technical Support Assistance

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

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

Revision History

Revision 1.1, February 2026

Section	Change Summary
Introduction	Updated section content to add details related to communication protocols.
Directory Structure and Files	Updated section content including Table 2.1. Folder Description.
Running the HMI Demo Design on CrossLinkU-NX SoM HW Board	• Updated Requirements section content including minor editorial fixes. • Updated Connecting the RPi CM5's Ethernet section content including updating steps and figures in sub-sections and adding additional sections in Setting Up Direct Connection to Host PC.
Debug Methodology	• Updated Figure 7.1. Debug Methodology Flow Chart. • Updated step 1 to add bullet points in Verifying Ethernet Connection. • Moved Programming the Bit File section here (previously in Running the HMI Demo Design on CrossLinkU-NX SoM HW Board). • Moved the Flashing the RPi CM5 section here (previously in Running the HMI Demo Design on CrossLinkU-NX SoM HW Board).

Revision 1.0, December 2025

Section	Change Summary
All	Initial release.

www.latticesemi.com