Lattice Sentry 4.0 MachXO5-NX LFMXO5-55TD Walkthrough Guide

User Guide

FPGA-UG-02217-1.0

July 2024

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

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviations	Definition
AES	Advanced Encryption Standard
AHB-Lite	Advanced High-performance Bus – Lite
APB	Advanced Peripheral Bus
AXI4	Advanced eXtensible Interface 4
BMC	Baseboard Management Controller
BSP	Board Support Package
CRE	Cryptographic Engine
ECDSA	Elliptic Curve Digital Signature Algorithm
ESFB	Embedded Security Function Block
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HDL	Hardware Description Language
I2C	Inter-Integrated Circuit
I3C	MIPI Improved Inter-Integrated Circuit
IP	Intellectual Property
ISK	Image Signing Key
JTAG	Joint Test Action Group
KAK	Key Authentication Key
LED	Light Emitting Diode
MRK	Master Root Key
OSC	Oscillator
OTP	One Time Programmable
PFR	Platform Firmware Resilience
PLL	Phase-Locked Loop
QSPI	Quad Serial Peripheral Interface
RBP	Rollback Protection
RISC-V	Reduced Instruction Set Computer-V (five)
RoT	Root of Trust
RX	Real Time OS (RISC-V for RTOS applications)
SDK	Software Development Kit
SGE	Software Generator Engine
SGPIO	Serial GPIO
SH-Bash	Shell Program
SKP	Secure Key Provisioning
SMBus	System Management Bus
SoC	System on Chip
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
UFM	User Flash Memory
USB	Universal Serial Bus

1. Introduction

This document provides a comprehensive guide on how to set up and use the Lattice Sentry™ 4.0 solution utilizing the MachXO5™-NX LFMXO5-55TD device. It walks you through the required tools, software set-up procedures, project workflow, programming and configuration, using demo and demo tools, and troubleshooting.

2. Required Tools

2.1. Software Requirements

- Lattice Propel™ Configuration:
  Lattice Propel: Propel2024.1_2406150513_p.exe
  Lattice Sentry Patch: Lattice_Sentry_4.0_Solution_(Propel2024.1 patch)_Prod.exe
- Lattice Radiant™ Configuration:
  Lattice Radiant: 2024.1.0.34.1_Radiant.exe
  Lattice Radiant Control Pack XO5D: 2024.1.0.34.1_Radiant_Ctrl_Pack_XO5D.exe
- Provision Tool
- Image Signing Tool
- Terminal emulator program such as PuTTY to view UART messages

Lattice Propel Design Environment, Lattice Radiant Software, and the Lattice Radiant Control Pack are available at https://www.latticesemi.com/.

Contact a Lattice representative to access the Provision Tool, Image Signing Tool, and Propel Sentry Patch.

2.2. Hardware Requirements

- Lattice Sentry Demo Board for MachXO5™-NX LFMXO5-55TD, Rev B
- 12 V power supply for the Lattice Sentry Demo Board
- 2 USB cables (Type-A to Type-B Mini)
- Two-position shunt connectors
- Lattice 2B Programming Cable
- DediProg programmer (optional, could be used for programming BMC image into an external flash)

2.3. License Requirements

The following licenses are required:

- LSC_CTL_LFMXO5-55TD
- LSC_ESFB55D_ES

3. One-time Software Setup Steps

3.1. Download IP from IP Catalog

In Lattice Propel Builder, make sure the following IPs are installed. Some IPs are installed automatically with the Lattice Sentry 4.0 Propel Patch. The rest can be downloaded from the IP Catalog. Make sure the version number of each IP matches the one shown in Table 3.1 to avoid potential compatibility issues. Figure 3.1 shows the IP Components Used in Sentry 4.0 Template.

Table 3.1. Sentry 4.0 Required IP

IP Name	Version	Source
GPIO	1.6.2	IP Catalog
RISC-V RX	2.4.0	IP Catalog
Lattice Sentry I2C Filter	1.3.0	IP Catalog
I3C Controller	3.3.0	IP Catalog
AXI4 to AHB-Lite Bridge	1.2.0	IP Catalog
AXI4 to APB Bridge	1.2.0	IP Catalog
AXI4 Interconnect	1.2.2	IP Catalog
AXI4-Lite Clock Management	1.1.0	Sentry 4.0 Propel Patch
Lattice RoT ESFB	2.0.0	Sentry 4.0 Propel Patch
Lattice Sentry QSPI Monitor	2.0.0	Sentry 4.0 Propel Patch
Lattice Sentry QSPI Master Streamer	2.0.0	Sentry 4.0 Propel Patch
SGPIO Master	1.1.0	Sentry 4.0 Propel Patch
Lattice Sentry SMBus Mailbox	2.2.0	Sentry 4.0 Propel Patch
Tightly Coupled Memory	1.4.0	Propel Base Install
AHB-Lite to APB Bridge	1.1.0	Propel Base Install
AHB-Lite Interconnect	1.3.1	Propel Base Install
APB-Interconnect	1.2.1	Propel Base Install
PLL	1.9.0	Propel Base Install
OSC for CRE	1.4.0	Propel Base Install

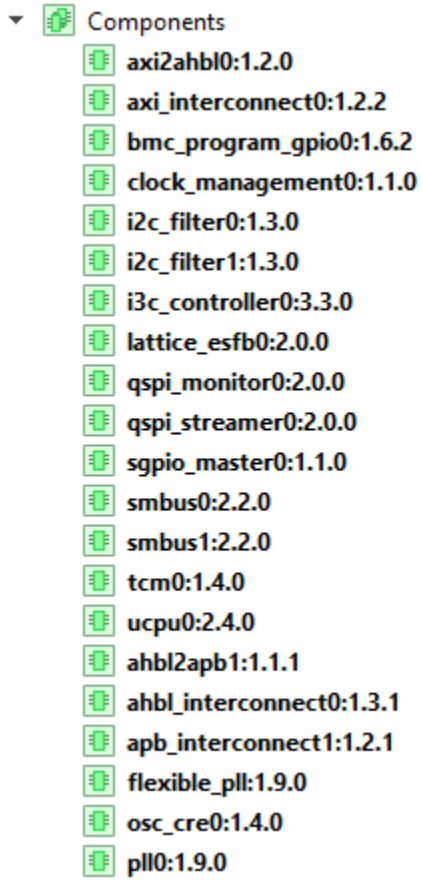

Figure 3.1. IP Components Used in Sentry 4.0 Template

3.2. Enable Controlled Device License in Lattice Radiant Software

To enable the controlled device license in Lattice Radiant software:

1. In Lattice Radiant software, from the menu, select **Tools** -> **Options** -> **Startup**.
2. Enable the Check Controlled Device License item (Figure 3.2).
3. Restart Lattice Radiant software for this setting to take effect.
4. Note that this setting should persist once it has been set.

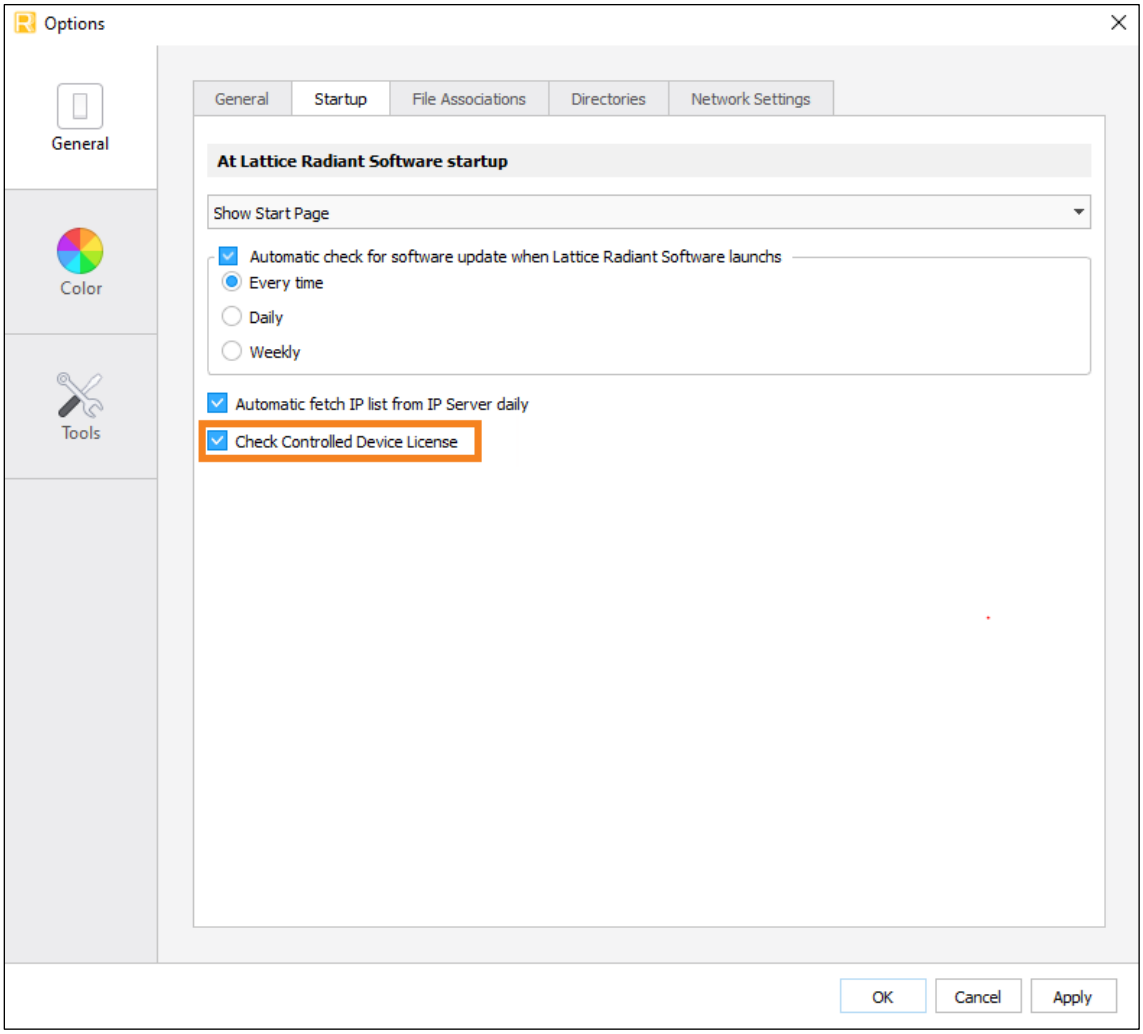

Figure 3.2. Check Controlled Device License

4. SoC and C Project Setup from Template

4.1. Create SoC Project in Lattice Propel Builder

Before creating the SoC project in Lattice Propel Builder, make sure you have downloaded the correct versions of all IP blocks used in the project, as shown in the Download IP from IP Catalog section.

1. Create a project directory for the SoC and C projects locally on your computer, as shown in Figure 4.1.

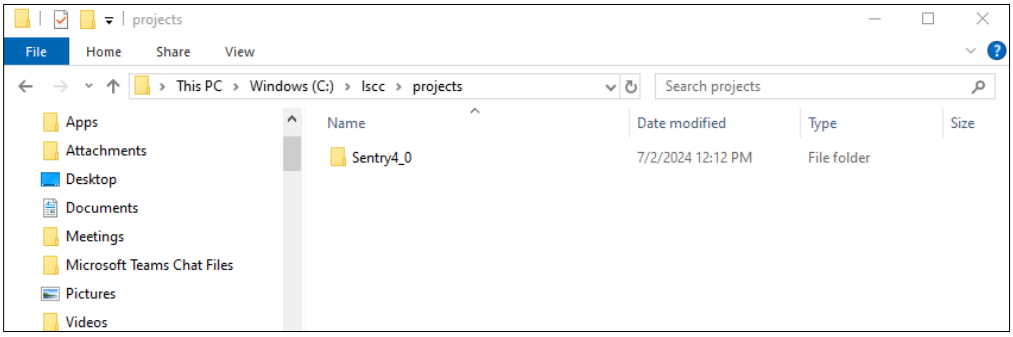

Figure 4.1. Local Project Directory for Creating SoC and C Project

2. Launch Lattice Propel Builder 2024.1.
3. From the menu, select **File** -> **New SoC Design**.
4. Enter a project name which ends in *_soc*. Click **Browse** to navigate to the empty project directory created in Step 1. Click **Next** (Figure 4.2).

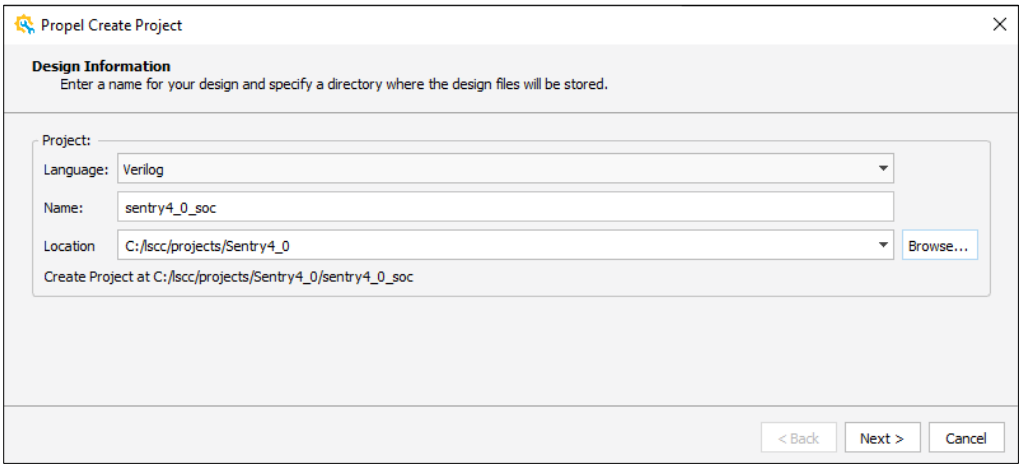

Figure 4.2. Design Information

5. Select Lattice Sentry 4.0 RoT Project, then click **Next** (Figure 4.3).

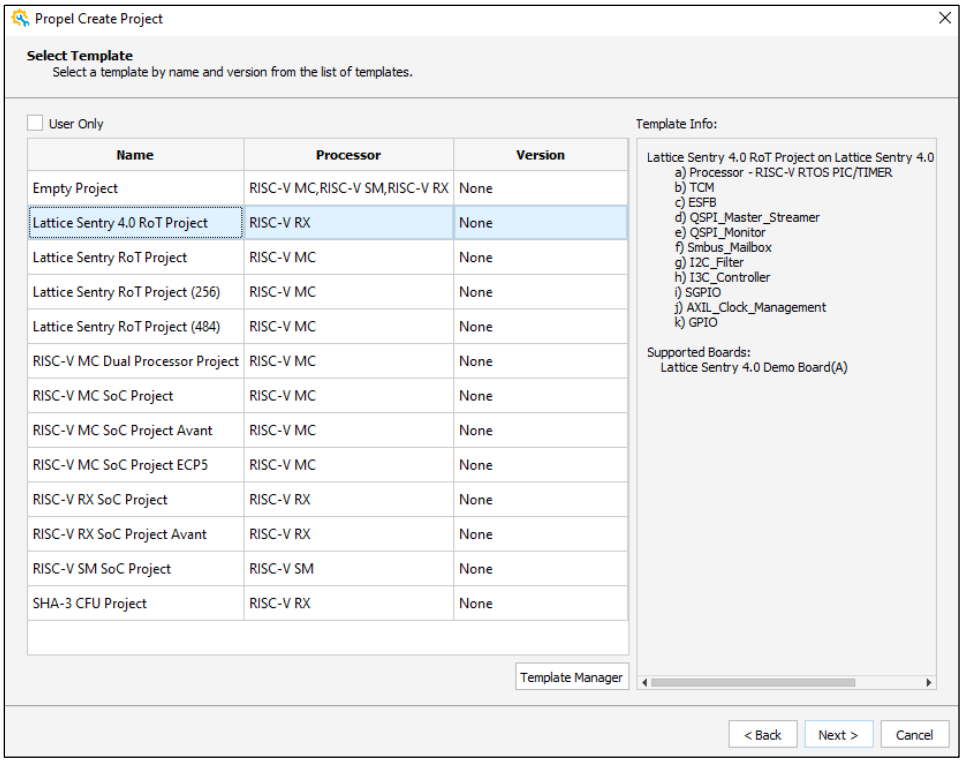

Figure 4.3. Create Lattice Sentry 4.0 RoT Project

6. Select Lattice Sentry 4.0 Demo Board for **Board**. Then, click **Next** (Figure 4.4).

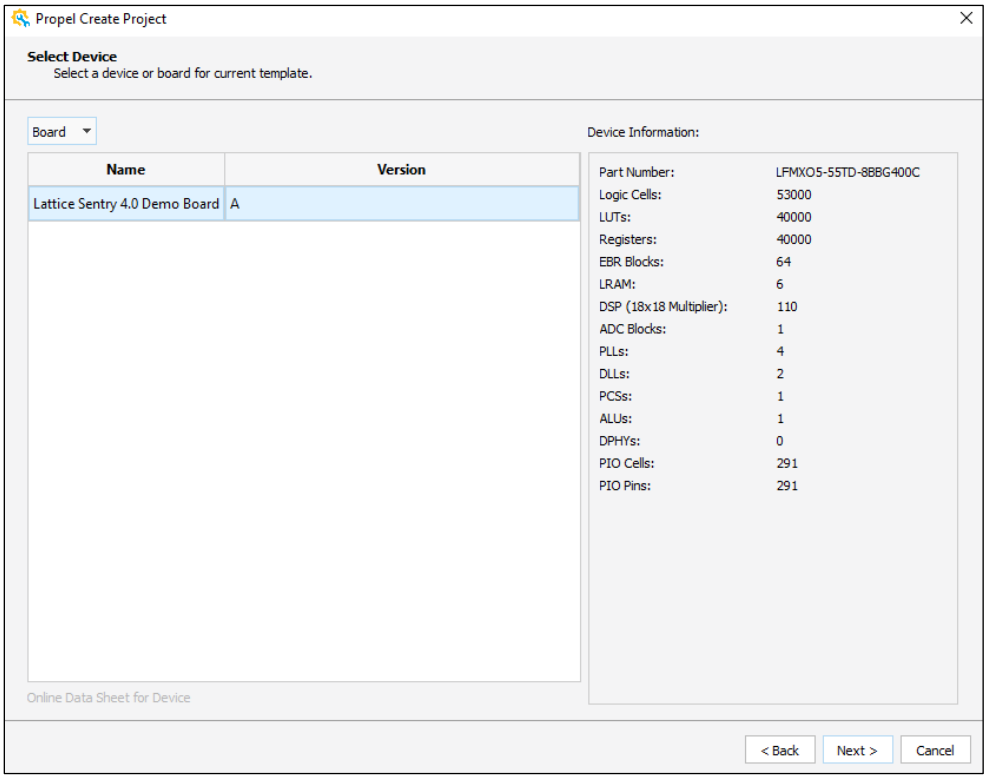

Figure 4.4. Create SoC Project – Select Device

7. Make sure the solution is Sentry 4.0. Then, click **Next** (Figure 4.5).

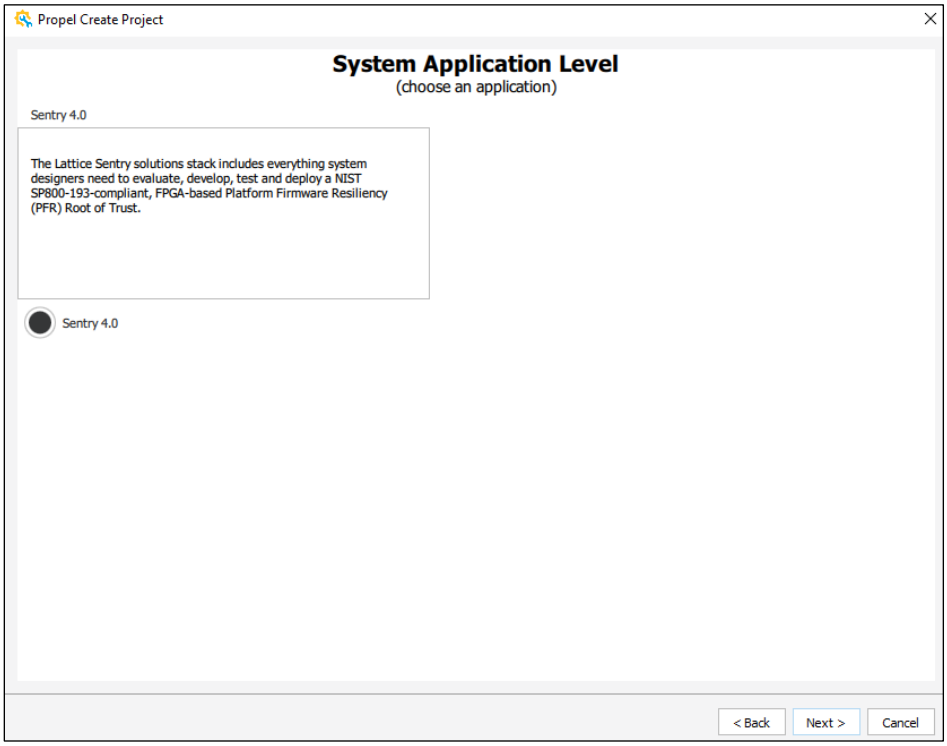

Figure 4.5. Create SoC Project – Select Application

8. Choose number of Peripherals, as shown in Figure 4.6. Then, click **Next**.

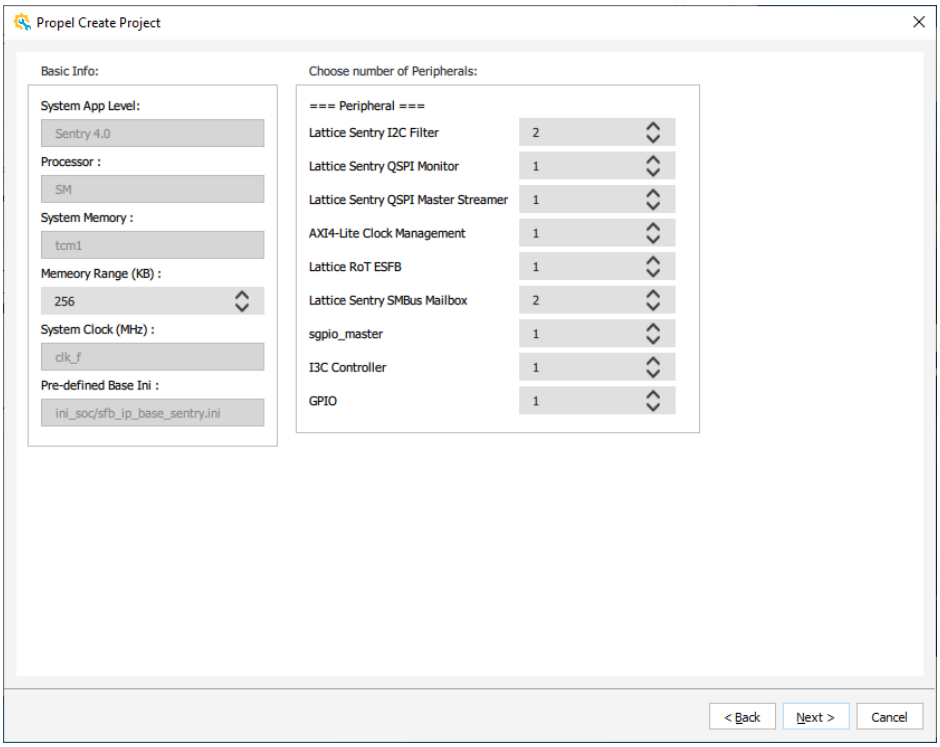

Figure 4.6. Create SoC Project – Select Peripherals

9. The next dialog box shows the SoC sketch. Click **Next** and finish (Figure 4.7).

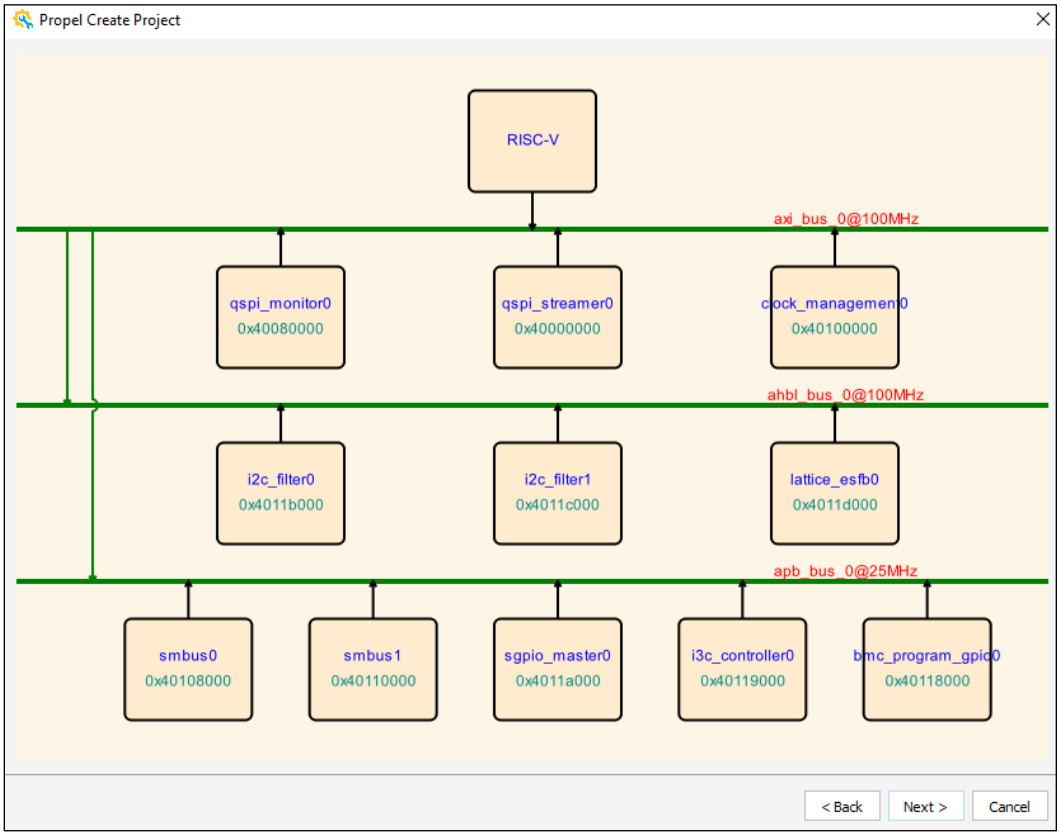

Figure 4.7. Create SoC Project – SoC Sketch

10. The SoC project opens in Lattice Propel Builder (Figure 4.8).

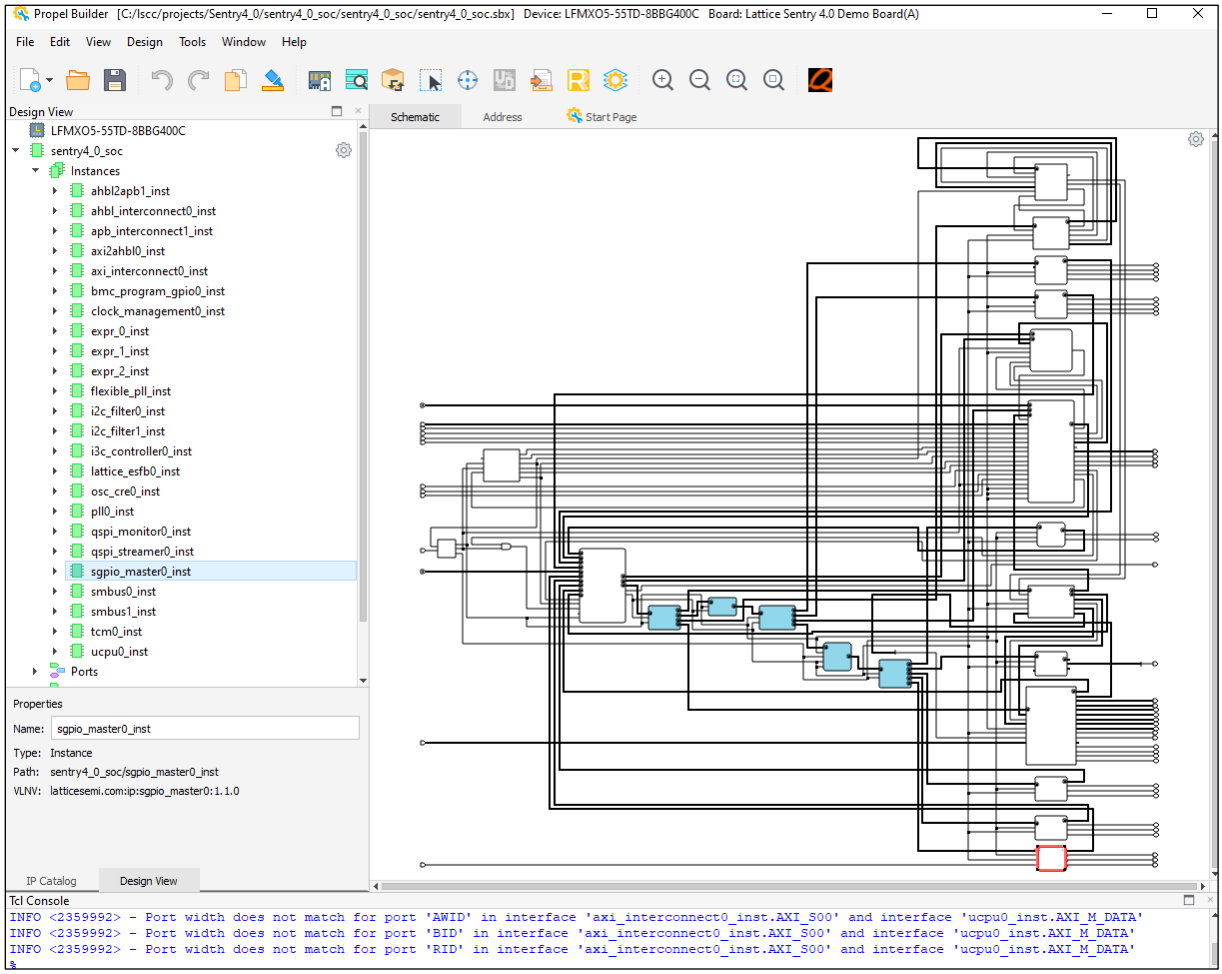

Figure 4.8. SoC Project – Created

11. Choose **File** -> **Save** or click the Save icon to save SoC project (Figure 4.9).

Figure 4.9. Propel Builder Save Icon

12. Choose **Design** -> **Validate** or click the Validate icon to validate SoC project (Figure 4.10).

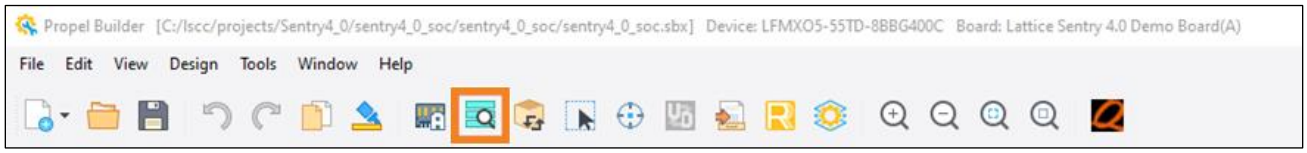

Figure 4.10. Propel Builder Validate Icon

13. Choose **Design** -> **Generate** or click the Generate icon to generate the SoC project (Figure 4.11).

Figure 4.11. Propel Builder Generate Icon

14. The SGE file, sys_env.xml, is generated through the Generate step. It is located in the following director:
    < Workspace> / <SoC Project> /sge/< sys_env.xml

4.2. Generate Bitstream in Lattice Radiant Software

1. In Lattice Propel Builder, choose **Design** -> **Run Radiant** or click the Lattice Radiant software icon to open the SoC project in Lattice Radiant software (Figure 4.12).

Figure 4.12. Lattice Radiant Software Icon

2. Sign the bitstream through the following steps.
   a. From the menu, choose **Tools** -> **Bitstream Security Settings**.
   b. When the Password dialog box pops up, click **OK** with the default password LATTICESEMI.
   c. In the Keys dialog box, make the following selections (Figure 4.13).
      - Check the checkbox next to **AES2 Encryption and enter key**:
        1F1316A755086123B563EC726DD9A5BA915C65BAC6584BF8E51C9A5ED37BF1F6
      - Check the checkbox next to **ECDSA**. Choose Authentication Mode ECDSA-256 from the drop-down menu and enter keys:
        Public Key
        6BD2BE7740EC2EBFF5AFB698296FFF3D09731CAC23C12E196BA92B5DADB65073CE0D320B4213D7ADD3F27C08B0158CBE9F0EC8DD023CAA9AFECA9EFD41FF49D7
        Private Key
        968411DA7986DC4F6031F84A641610E2DB6A4C3BDA1550B11E619A51266A7CB2
      - For **KeyBlob**, choose Normal from the drop-down menu. Choose Authentication Mode ECDSA-384 from the drop-down menu.
      - For **KAK**, enter keys:
        Public Key
        1f6c4a9fddcf7fbd815f4e9bcc0a497b54f222dda9bdc1c3bd1a79ea283bb2ff4288639f34c173c0acf8456c0be757a0d5ff99832c676033a772979b738fd8bacd3975c3df48cfd996ceb948698ae23c23d6cee67bd07d4b7dbd147977071258
        Private Key
        9223bae8b8ce2fe76847f1a7a82651217d3f5698656658a3cb43cf7369c80a8f85a898bbf046c87259395e89921e60a4
      - For **ISK**, enter ISK ID 0.
      - Click Auto Generated Key Pair for ISK.
        Note: The ISK key pairs are automatically saved in the Security Settings folder of the SoC project. Once generated, these keys can be used consistently during project development, until an ISK key is revoked. This eliminates the need to constantly auto-generate new ISK key pairs.
   d. Click **OK**.

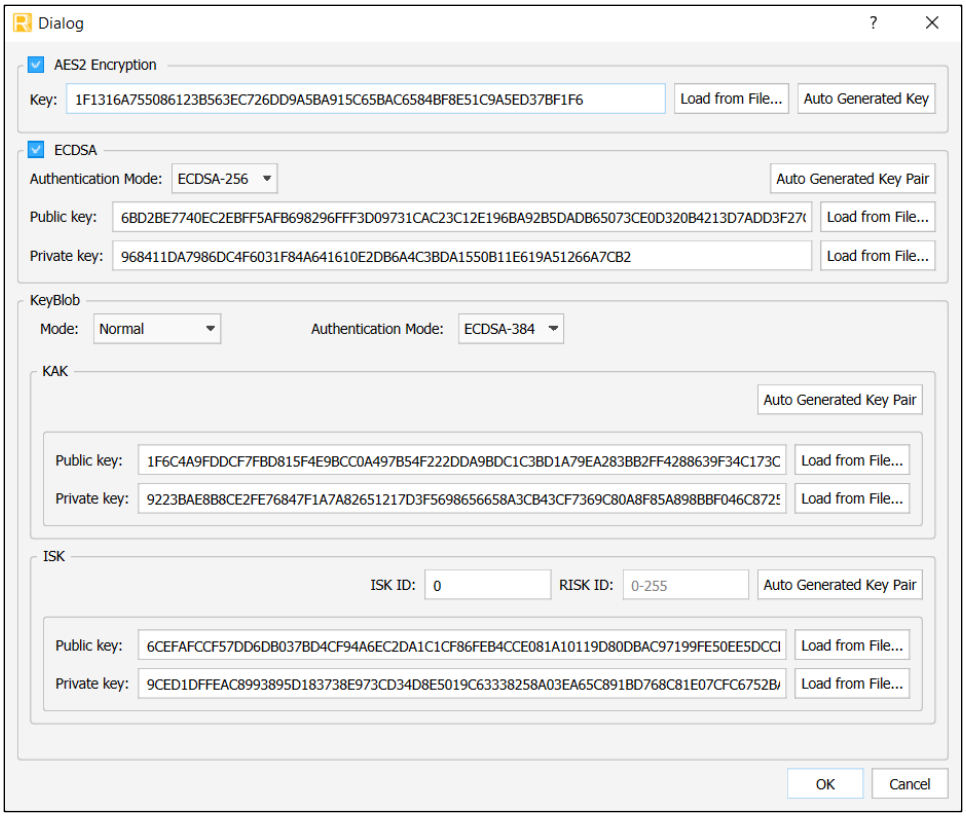

Figure 4.13. Key Dialog Box

3. After you click **OK** in this dialog box, the keyblob and keys are stored in a Security Setting directory in the following location: < Workspace>/<SoC Project>/security_setting (Figure 4.14).

   The files keyblob_normal.bin and isk.prv are input to the image signing tool, which is described in the Sign the Firmware File section. These files and other key files may be saved and used during project development. Key files can also be loaded directly into Lattice Radiant Bitstream Security Settings without the need of copy-pasting keys into the text boxes.

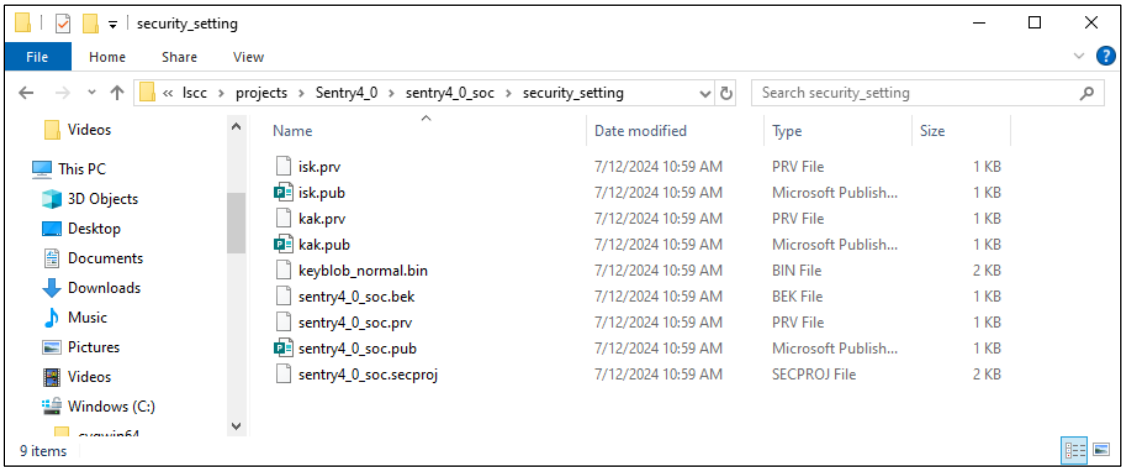

Figure 4.14. Security Setting Directory

4. In Lattice Radiant software, click the green arrow to run through the Synthesis/Map/PAR/export process to generate a signed bitstream (Figure 4.15).

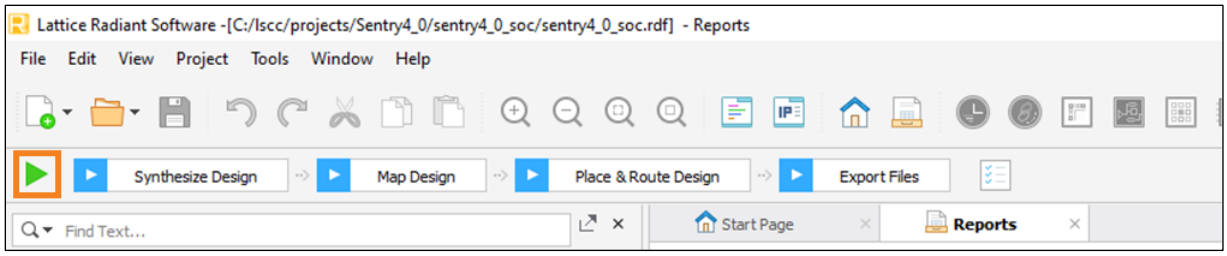

Figure 4.15. Lattice Radiant Software Synthesis/Map/PAR Process

5. The generated bitstream is located in the following directory:
   < Workspace>/<SoC Project>/impl1/<project_name>_impl_1.bit

4.3. Create C Project in Lattice Propel SDK

1. In Lattice Propel Builder, choose **Design** -> **Run Propel SDK** or click the Lattice Propel SDK icon in the ribbon to create and open the firmware project (Figure 4.16).

Figure 4.16. Lattice Propel SDK Icon

2. In the Lattice Propel Launcher dialog box, click **Browse** and navigate to the directory created for the SoC and C projects in the Create SoC Project in Lattice Propel Builder section, step 1 (Figure 4.17).

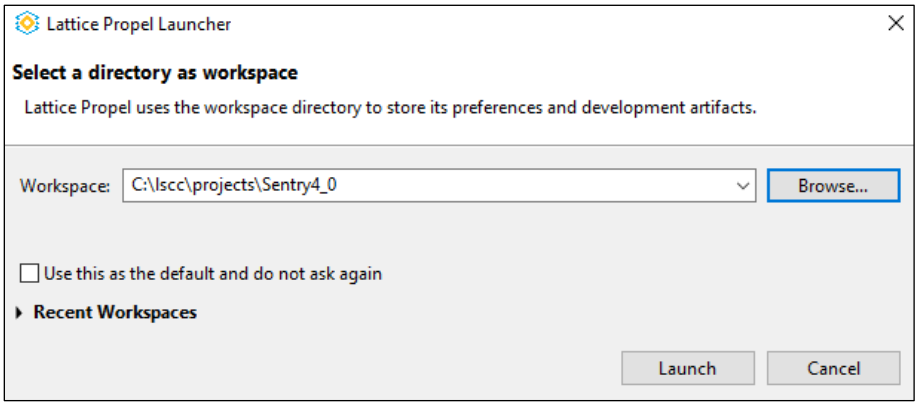

Figure 4.17. Lattice Propel Launcher – Select Directory for Creating SoC and C Project

3. Lattice Propel SDK is launched from Propel Builder. The tools automatically locates the sys_env.xml file, which is located in the sge folder of the SoC Project directory, and associate the C project with this SoC file. This ensures the addresses and IP drivers align. If the tools do not automatically locate the sys_env.xml file, click **Browse**, located next to the **System env** text box, and navigate to the sys_env.xml file manually.
4. Enter a project name for the firmware project and click **Next** (Figure 4.18).

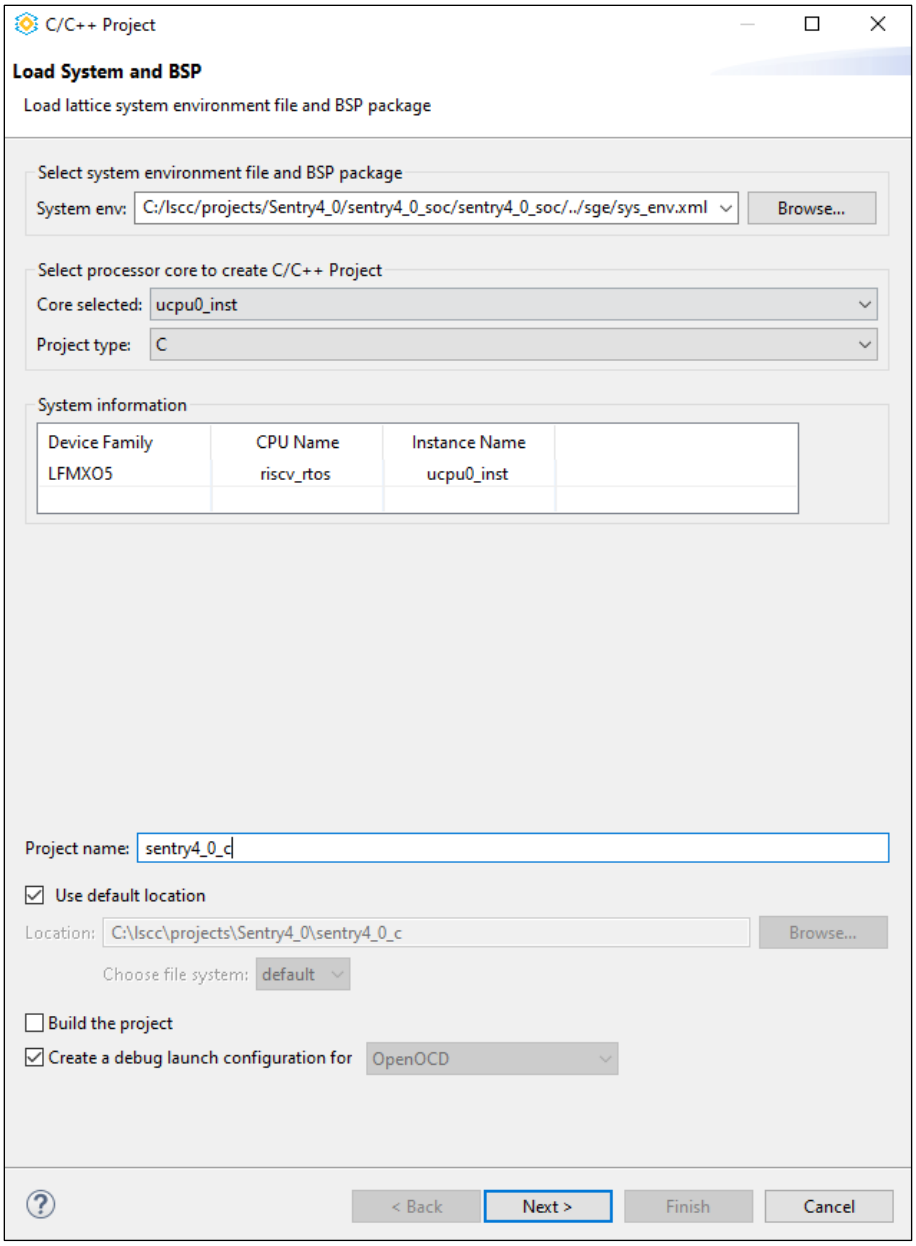

Figure 4.18. Create Firmware Project

5. In the following dialog box, keep the default options and click **Finish**.
6. The firmware project is generated in Lattice Propel SDK.
7. Select the firmware project from the Project Explorer view and click the Build icon to build it (Figure 4.19).

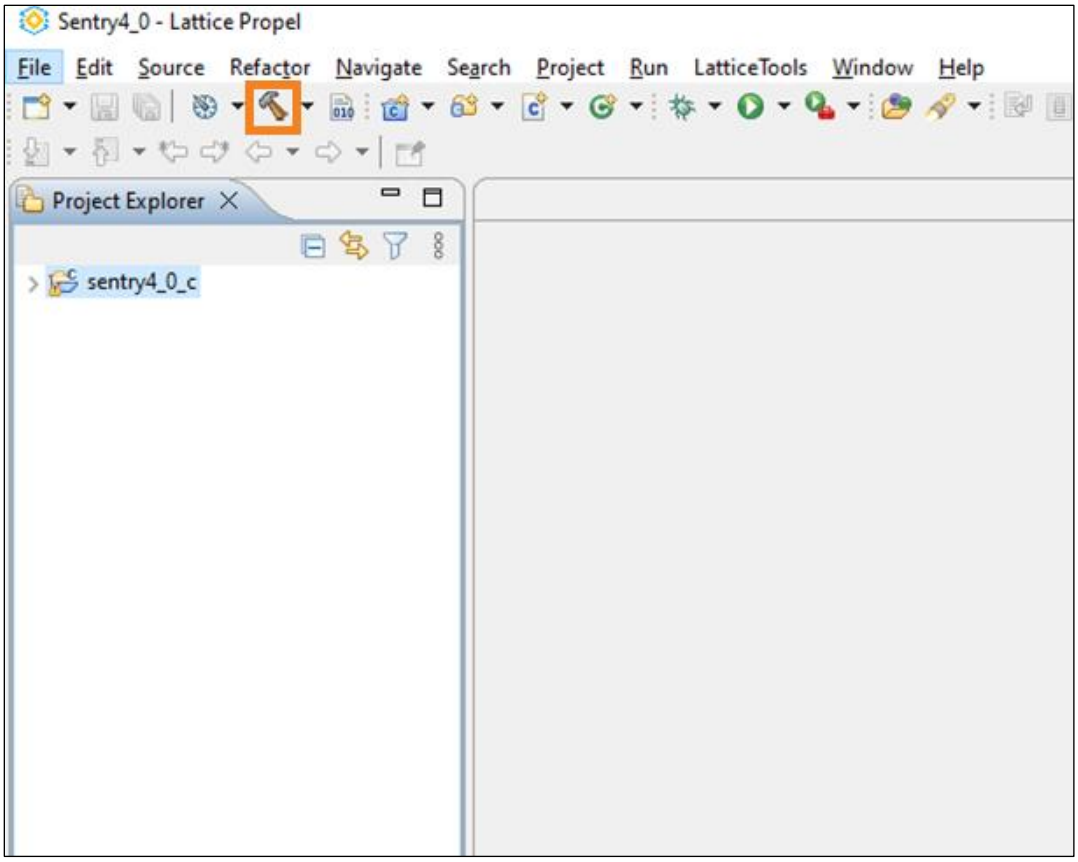

Figure 4.19. Build Firmware Project

8. The project should be built without errors.
9. The .mem file generated when the project is built is located in the following directory (Figure 4.20):
   < Workspace>/<C Project>/Debug/<project_name>.mem

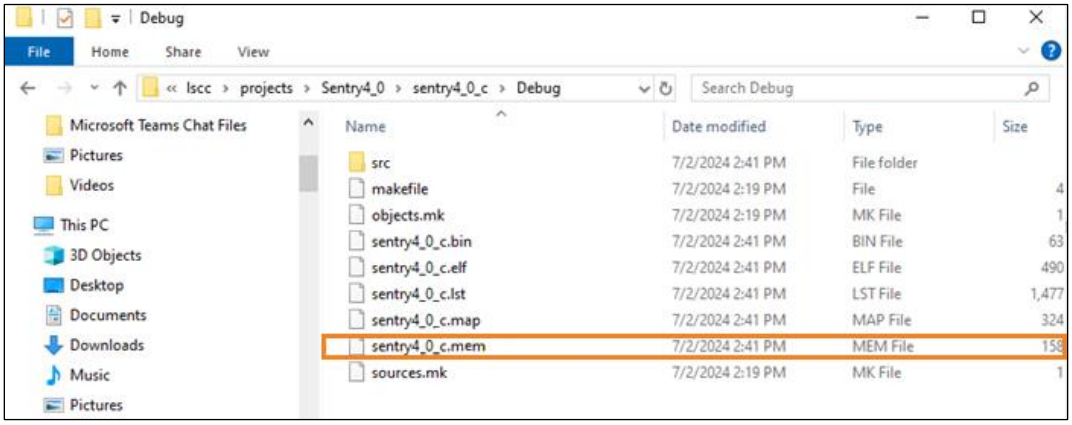

Figure 4.20. Location of Firmware .mem File

10. This .mem file needs to be signed using an image signing tool, before being programmed through the provisioning flow. See Sign the Firmware File and Program MachXO5-NX LFMXO5-55TD Device with the Provisioning Tool sections for more details.

5. Importing SoC and C Projects from Existing Project

This section describes the workflow to import existing SoC and Firmware projects into a fresh workspace. It should be used if one is working on a design which has components or firmware added on top of the base template.

5.1. Create a Blank Workspace

1. Create a project directory for the SoC and C projects locally on your computer (Figure 5.1).

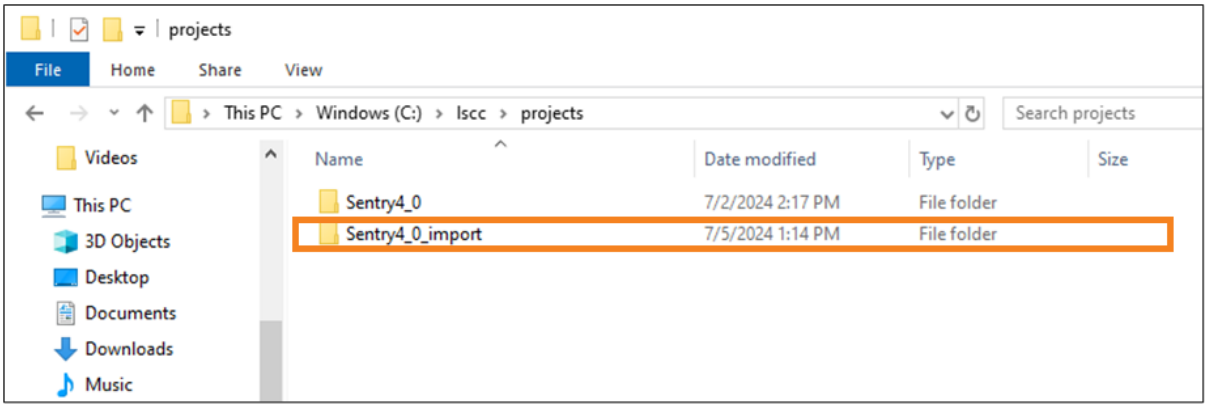

Figure 5.1. Project Directory for Importing SoC and C Projects

2. Ensure that the directory from which you are importing the SoC project is correct.

5.2. Import SoC Project into Workspace

1. Launch Lattice Propel SDK 2024.1.
2. In the Lattice Propel launcher, click **Browse** and navigate to the directory created in the Create a Blank Workspace section, Step 1 (Figure 5.2).

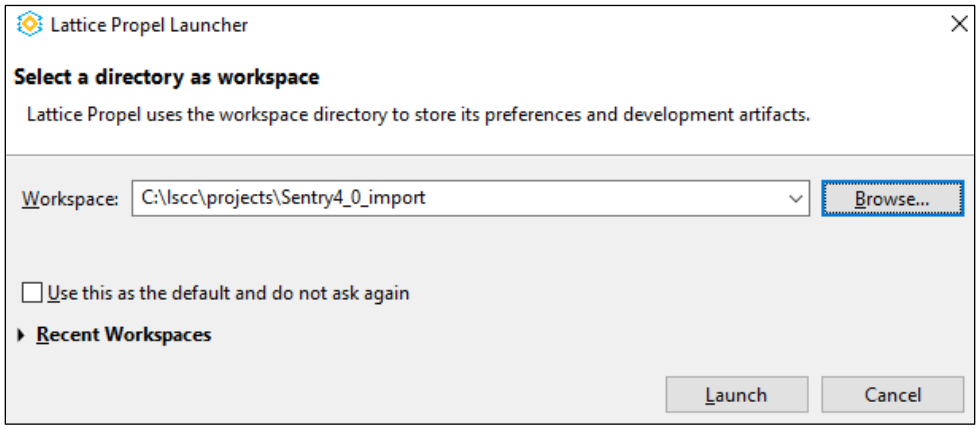

Figure 5.2. Lattice Propel Launcher – Select Directory for Importing SoC and C Projects

3. In the **Project Explorer** view, click **Import projects**, or choose **File** -> **Import Projects** from the menu (Figure 5.3).

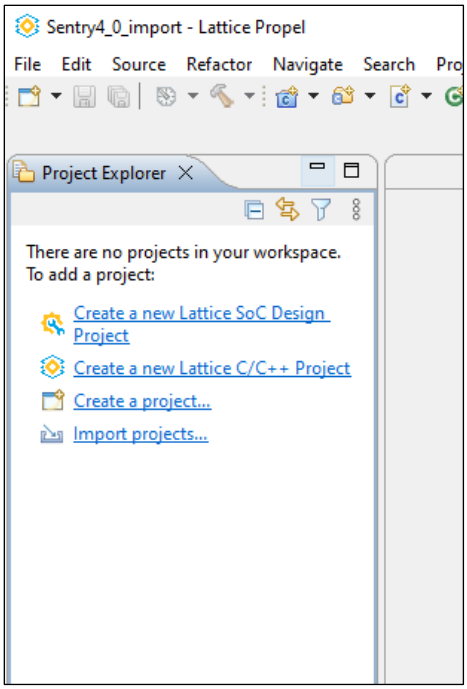

Figure 5.3. Import Projects

4. In the Import window, select **Lattice Propel** -> **Lattice SoC Design Projects into Workspace**. Click **Next** (Figure 5.4).

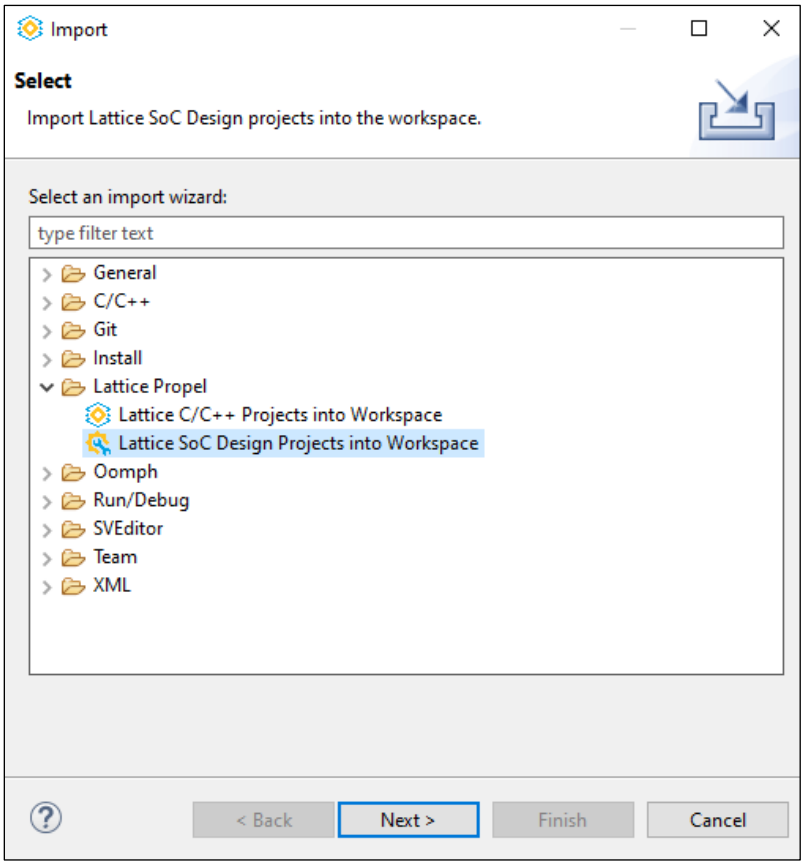

Figure 5.4. Import SoC Design Projects into Workspace

5. Click **Browse** next to **Select Root Directory** and navigate to the directory which contains the SoC project to be imported. Click **Select Folder** (Figure 5.5).

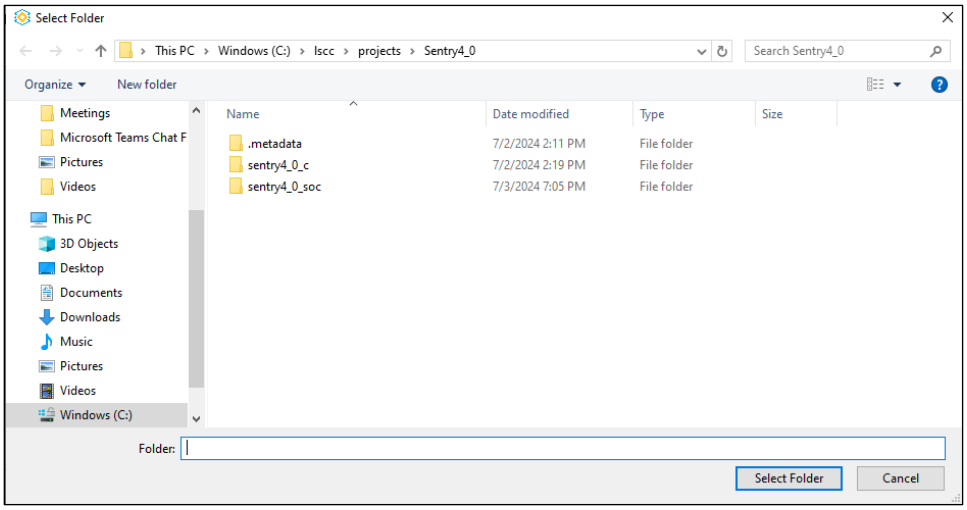

Figure 5.5. Workspace with SoC Project to Import

6. The Import tool automatically identifies the SoC project (Figure 5.6). If the SoC project does not show up in the list, make sure that the SoC project's outer directory folder ends with the characters "_soc," and repeat Steps 3-6.

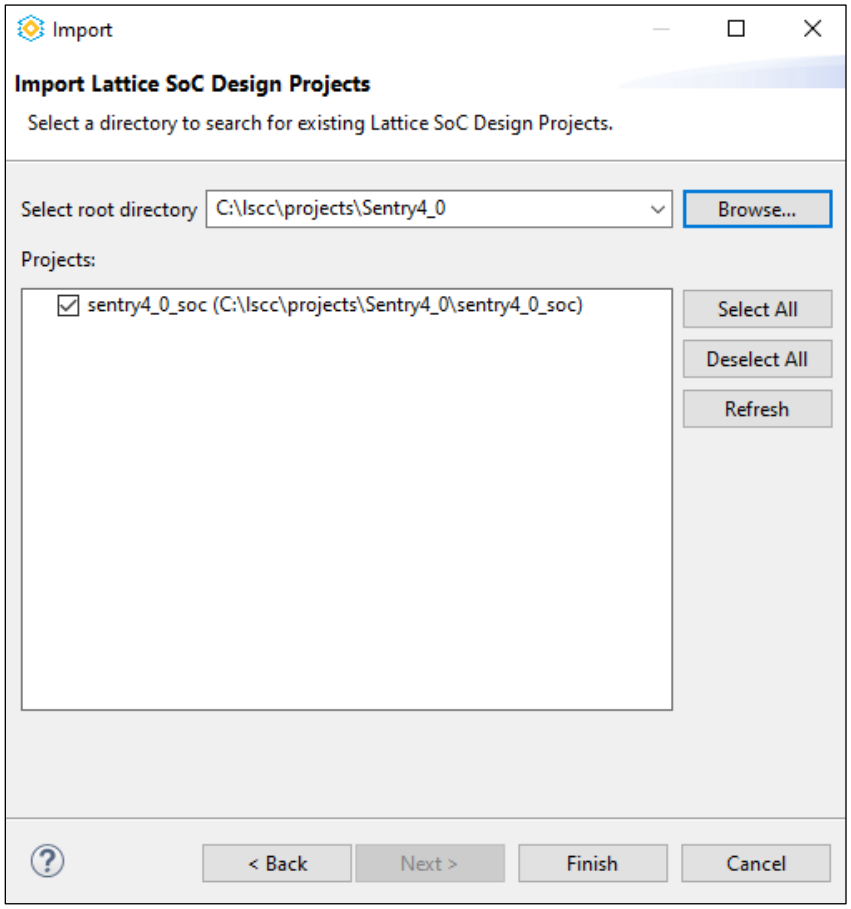

Figure 5.6. Identify Lattice SoC Design Projects

7. Select the SoC project and click **Finish**.
8. The SoC project directory shows up in the Project Explorer view.
9. Select the SoC project and click the Propel Builder icon to open the SoC project in Lattice Propel Builder.
10. The SoC project opens in Lattice Propel Builder (Figure 5.7).

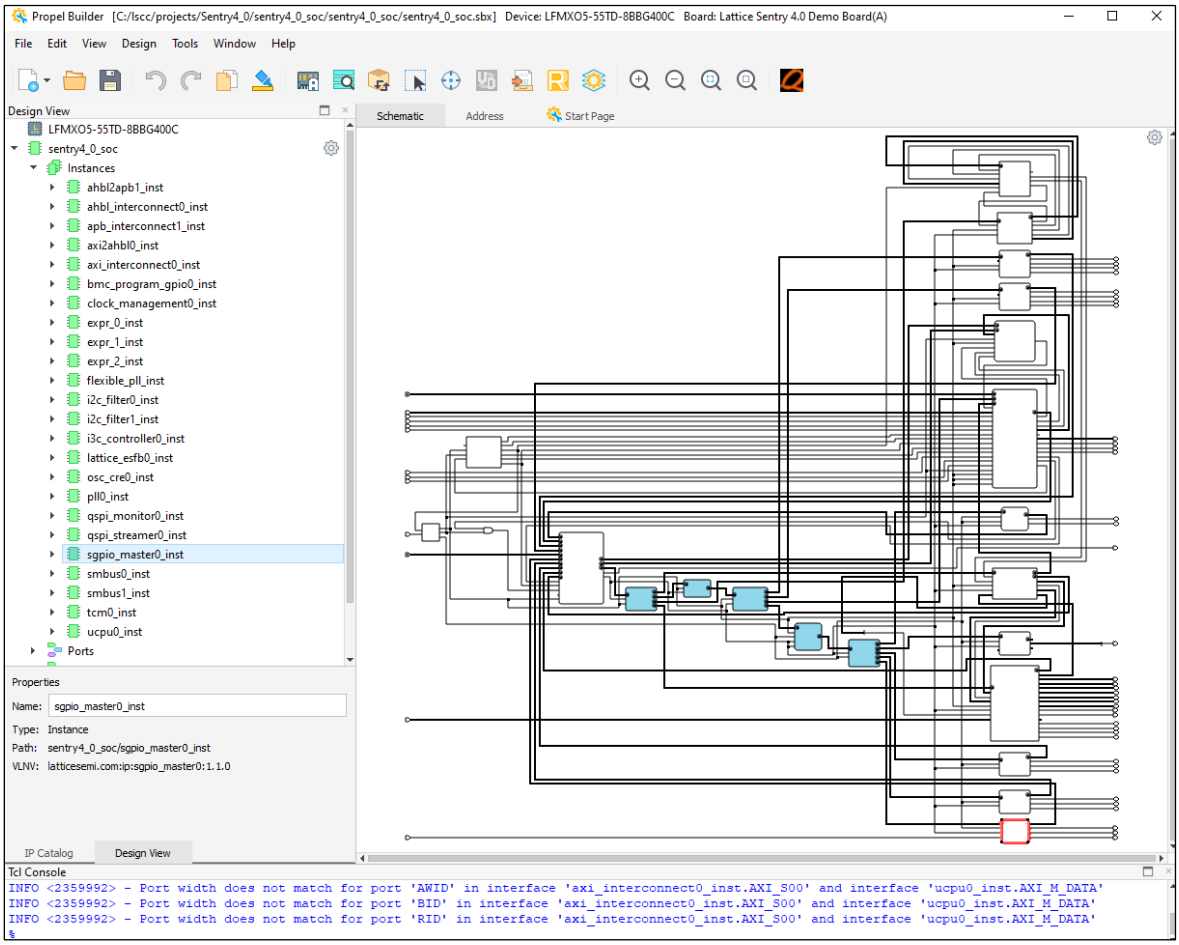

Figure 5.7. SoC Project – Imported

11. Choose **File** -> **Save** or click the Save icon to save the SoC project (Figure 4.9).
12. Choose **Design** -> **Validate** or click the Validate icon to validate SoC project (Figure 4.10).
13. Choose **Design** -> **Generate** or click the Generate icon to generate the SoC project (Figure 4.11).
14. Follow the steps in the Generate Bitstream in Lattice Radiant Software section to open the SoC project in Lattice Radiant and generate a bitstream.

5.3. Import C Project into Workspace

1. Launch Lattice Propel SDK 2024.1.
2. In Lattice Propel Launcher, click **Browse** and navigate to the directory created in the Create a Blank Workspace section, Step 1 (Figure 5.2).
3. In the Project Explorer view, click **Import projects**, or choose **File** -> **Import Projects** from the menu (Figure 5.3).
4. In the Import window, select **Lattice Propel** -> **Lattice C/C++ Projects Into Workspace**. Click **Next** (Figure 5.8).

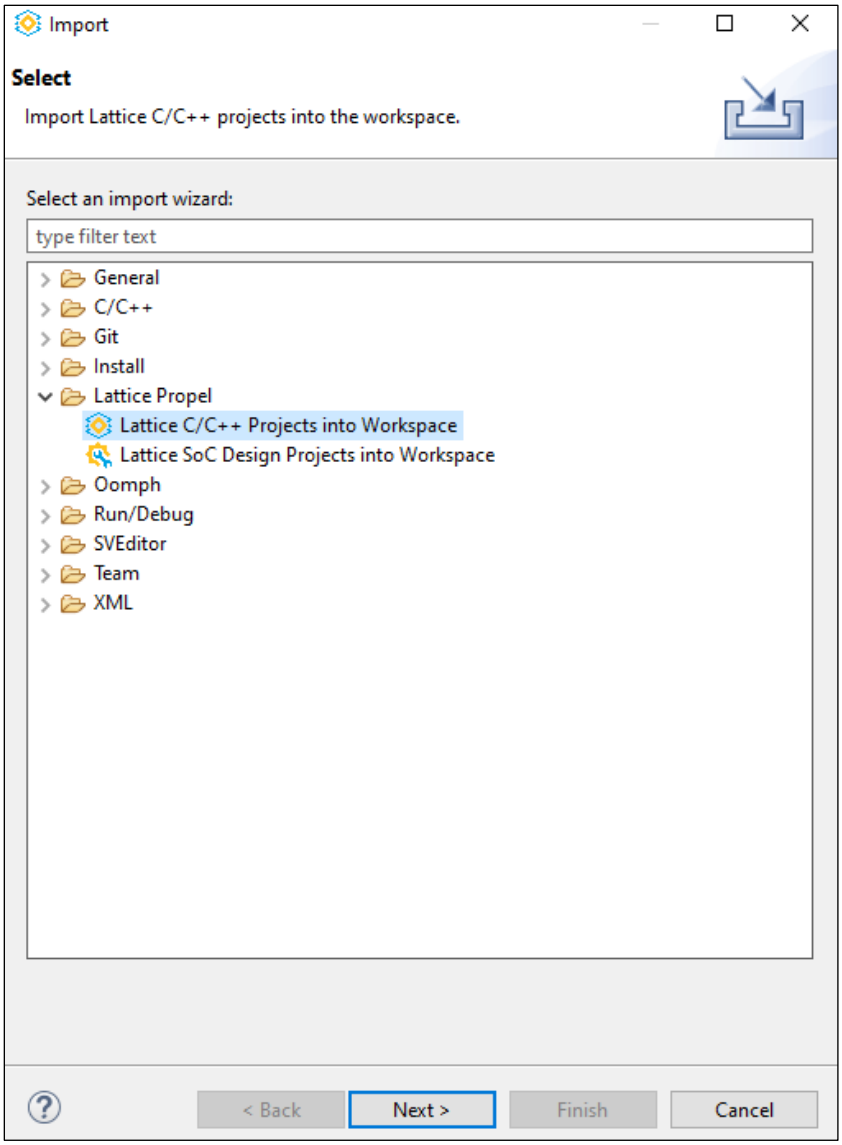

Figure 5.8. Import Lattice C/C++ Projects into Workspace

5. Click **Browse** next to **Select Root Directory** and navigate to the directory which contains the C project to be imported. Click **Select Folder** (Figure 5.9).

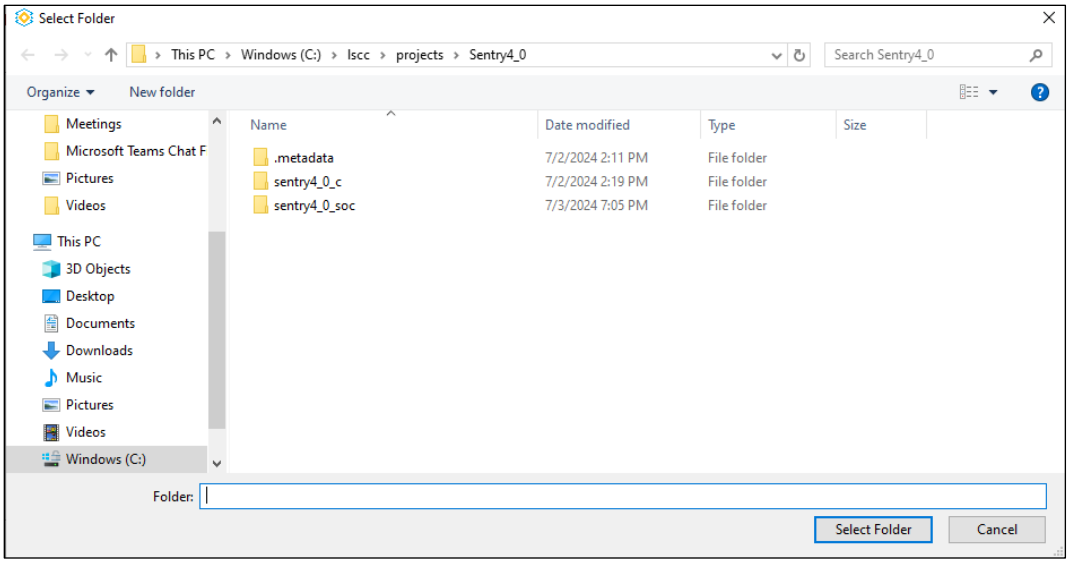

Figure 5.9. Import C Project into Workspace

6. The Import tool automatically identifies the C project, as shown in Figure 5.10.

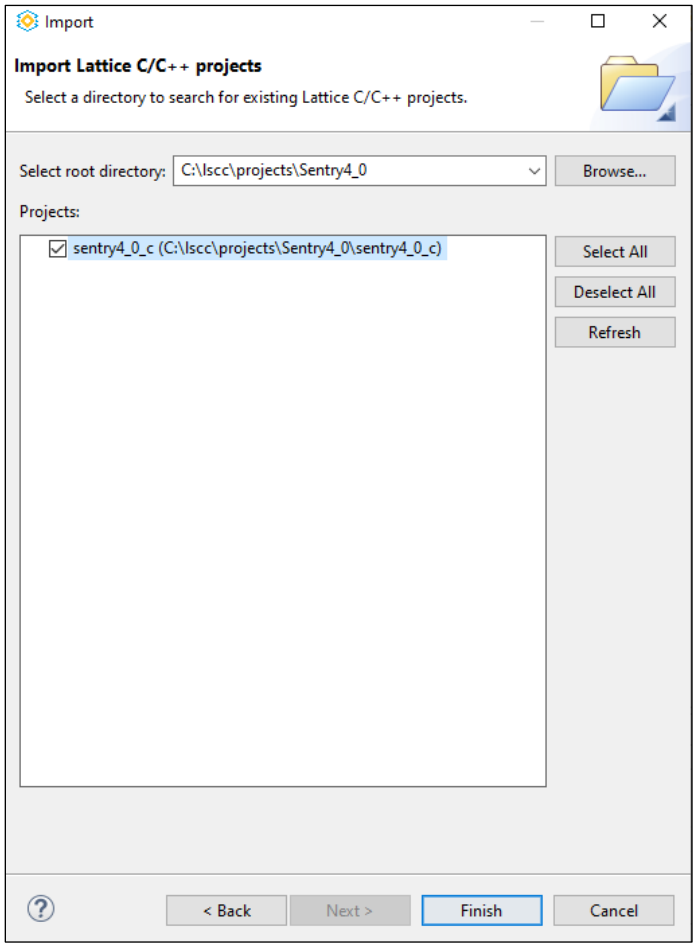

Figure 5.10. Identify Lattice C/C++ Projects

7. Select the C project and click **Finish**.
8. When a dialog box pops up to ask about overwriting .setting, click **Yes to All** (Figure 5.11).

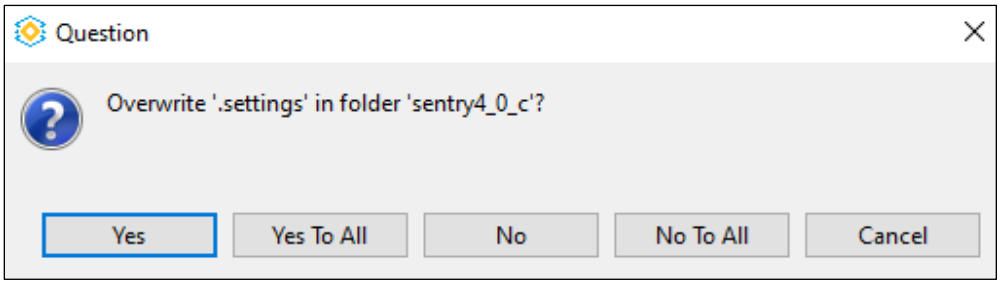

Figure 5.11. C Project Import Settings Question

9. The C project directory shows up in the Project Explorer view.
10. Select the firmware project in the Project Explorer view and click the Build icon to build it (Figure 4.19).
11. The .mem file generated when the project is built is located in the following directory (Figure 4.20):
    < Workspace>/<C Project>/Debug/<project_name> mem
12. This .mem file needs to be signed using an image signing tool, before being programmed through the provisioning flow. See the Sign the Firmware File and Program MachXO5-NX LFMXO5-55TD Device with the Provisioning Tool sections for more details.

6. Project Workflow

As changes are made to the C or SoC projects, this workflow must be followed to update other pieces of the project.

6.1. Lattice Propel SDK Workflow

If any updates are made to the C project, after building the project, the .mem file must be signed with an image signing tool and the device can be re-provisioned with the new firmware using the Provisioning Tool.

6.2. Lattice Propel Builder Workflow

If any updates are made to the SoC project in Lattice Propel Builder, the project must be re-generated in Lattice Propel Builder. Then, Synthesis/Map/PAR/Export must be rerun in Radiant.

If modifications made to the SoC project in Lattice Propel Builder causes the memory map to change, such as adding or removing a component, the Lattice Propel SDK firmware project must be updated.

To update the Lattice Propel SDK firmware project:

1. In the Project Explorer view of Lattice Propel SDK, right-click on the C project and choose **Update Lattice C/C++ Project…** (Figure 6.1).

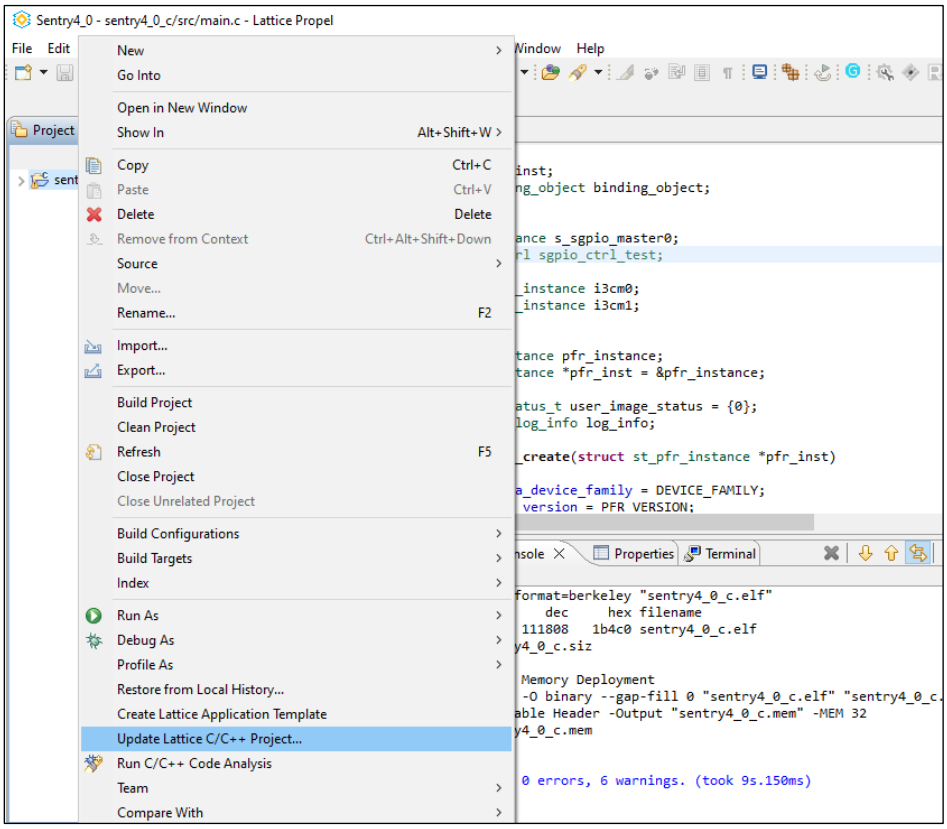

Figure 6.1. Update Lattice C Project

2. In the Update System and BSP dialog box, click **Browse** next to **New System Env** and navigate to the following file (Figure 6.2):

   < Workspace >/< SoC Project >/sge/sys_env.xml

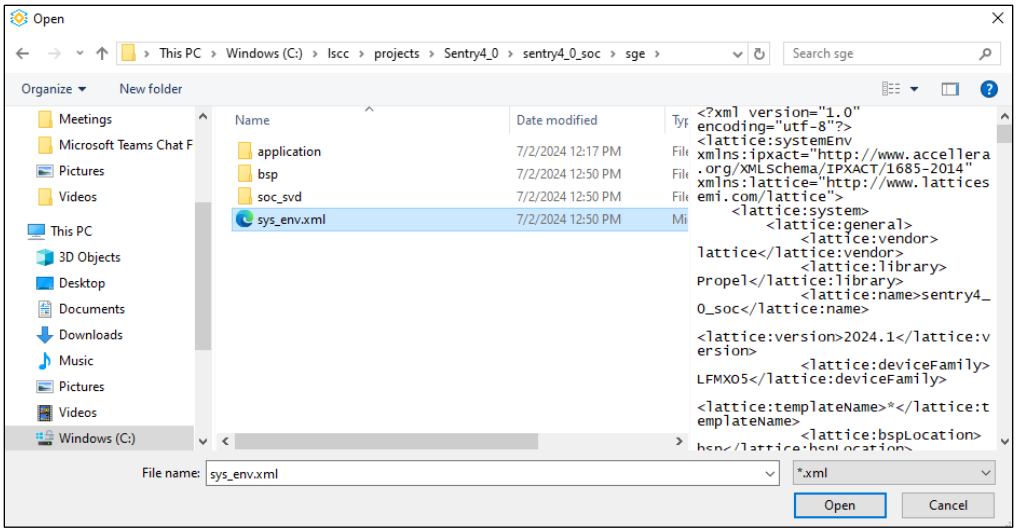

Figure 6.2. Directory Location of sys_env.xml

3. Click **Open**.
4. Check the checkbox next to **Re-generate toolchain parameters and linker script** (Figure 6.3).

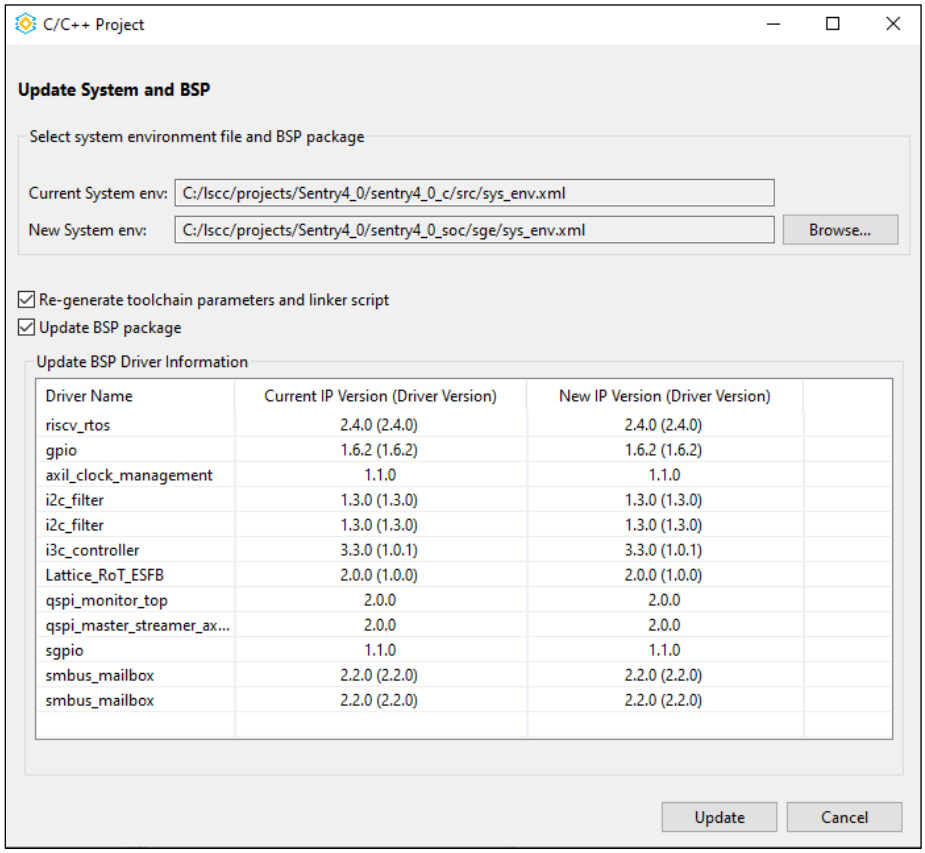

Figure 6.3. Update System and BSP

5. If the checkbox for Update BSP Package is checked, all firmware driver files in the BSP directory are replaced with the default firmware drivers provided by the IP Catalog. Any changes that have been made to any files in the BSP

directory are overwritten. Checking this checkbox also updates sys_platform.h with the addresses and other macros associated with the most recently generated Lattice Propel Builder SoC project.

6. If changes have been made to any files in the BSP directory, make a backup copy of these files before checking the checkbox for Update BSP Package. After updating the system and BSP, manually replace the default firmware driver files with the backup ones.
7. Click **Update**.
8. Now that the C project has been updated. Refer to Lattice Propel SDK Workflow.

6.3. Lattice Radiant Software Workflow

If modifications are made to the constraints or HDL code in Lattice Radiant software, the Synthesis/Map/PAR/export process must be rerun. This is the final step in the project build process. No updates to Lattice Propel SDK or Lattice Propel Builder are necessary.

7. Policy Editor

The Policy Editor is a GUI-based tool accessible from Lattice Propel SDK. It is used to set the customer keys, customer lock settings, and customer policy settings. It generates a customer_policy.bin file which is programmed into the MachXO5-NX LFMXO5-55TD device using the Provision Tool. A sample customer_policy.bin file is provided in the Provision Tool directory.

1. Launch Lattice Propel SDK and open the current workspace.
2. From the menu, select **Lattice Tools** -> **Lattice RoT Tools for MachXO5-55TD** -> **Policy Editor**, as shown in Figure 7.1.

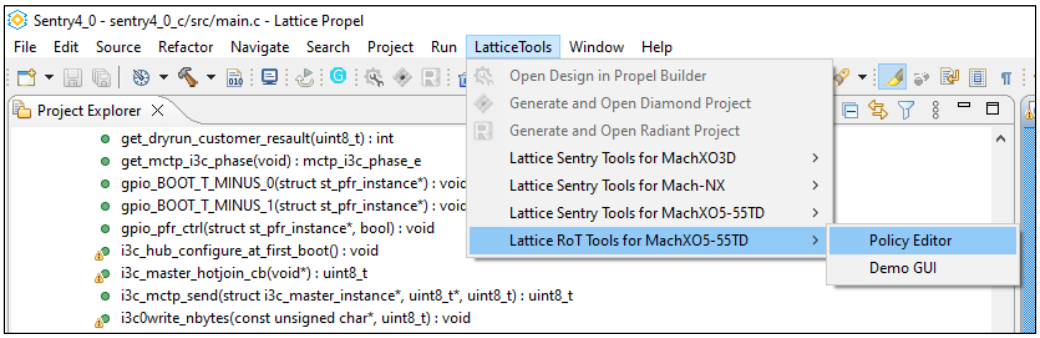

Figure 7.1. Open Policy Editor

3. The Policy Editor GUI opens (Figure 7.2).

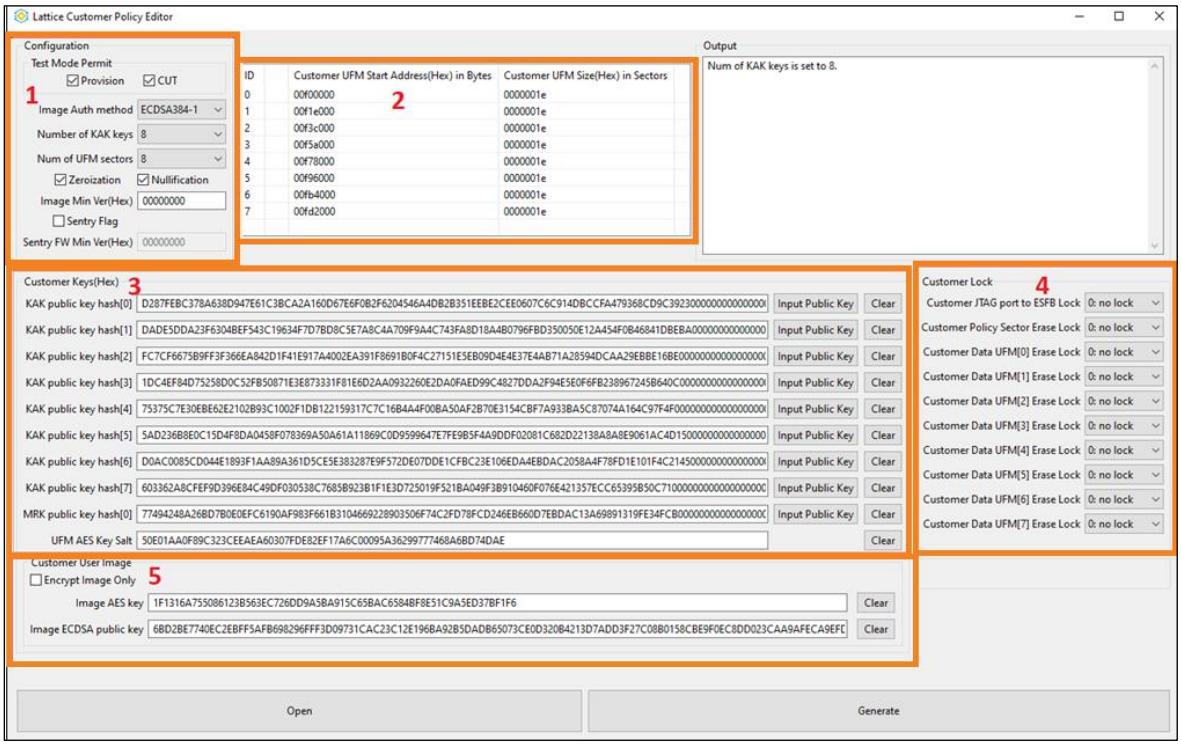

Figure 7.2. Policy Editor GUI

Policy Editor includes the following components, as labelled in Figure 7.2:

Area 1 for Configuration Settings, Area 2 for UFM Sector definitions, Area 3 for Customer Keys, Area 4 for Customer Lock, and Area 5 for Customer User Image. These components are listed below and introduced one by one.

- Configuration Settings (1)
  - Test Mode Permit:
    If both boxes are checked, the test mode pins (SW4 on the Lattice Sentry 4.0 Demo Board) can be used. If either box is unchecked, the test mode pins cannot be used to go into that mode, and the board always boots to Normal Mode. It is recommended to keep these boxes checked during development.
  - Image Auth Method:
    Authentication method for KAK, ISK, and MRK keys.
  - Number of KAK Keys:
    You can configure 1-8 KAK keys.
  - Number of UFM Sectors:
    You can configure 1-8 UFM sectors.
  - Zeroization/Nullification:
    Checking these boxes means that zeroization and/or nullification is allowed.
  - Image Min Ver:
    Minimum bootable SoC image version. Enter 00000000 to allow any SoC image to boot, regardless of version number. SoC image is set by Lattice Radiant software when the bitstream is generated. By default, it is the timestamp at the time of bitstream generation.
  - Sentry Flag:
    Sentry Flag must be checked.
  - Sentry FW Min Ver:
    Minimum bootable firmware version. Enter 00000000 to allow any firmware to boot, regardless of version number. The firmware version is an argument given to the image signing tool. If a signed firmware's version is lower than the Sentry FW Min Ver in the customer policy, the firmware does not boot.
- UFM Sectors (2)
  - There is a total of 1 MB reserved for User Flash Memory. You can configure this space however you choose, in 1-8 UFM sectors.
  - The minimum size for one UFM sector is 4 KB, and this unit is also referred to as a sector since it is essentially a SPI Sector.
  - General use UFM begins at address 0xF00000.
  - For every UFM sector, you can configure the starting address and select how many SPI sectors are included.
  - In the default case, all eight UFM sectors are used and they are of equal size, so each UFM sector contains 31 SPI sectors, or 128 KB.
- Customer Keys (3)
  - You can configure up to eight KAK keys. For each KAK key, you should enter the key and the Policy Editor GUI calculates the hash. Only the hash is stored on the device, not the key itself.
  - AES Key Salt:
    This AES key salt is used to encrypt the UFM content using the AES GCM algorithm.
- Customer Locks (4)
  - The customer JTAG port, customer policy sector, and each UFM sector can be unlocked, soft locked, or hard locked.
  - No lock:
    The port or sector is fully accessible.
  - Soft lock:
    The port or sector can be locked and unlocked by the Orchestration FW to perform updates via the ESFB.
  - Hard lock:
    The port or sector is permanently locked. The port is not accessible. The sector is OTP, read-only.

- Customer User Image Key (5)
  - The Image AES Key in the Policy Editor must be the same as the AES2 Encryption key used in Lattice Radiant software's Bitstream Security Settings, which is used to sign the customer image .bit file.
  - The Image ECDSA Public Key in the Policy Editor must be the same as the ECDSA Public Key used in Lattice Radiant software's Bitstream Security Settings, which is used to sign the customer image .bit file.
  - The image AES key and image ECDSA public key are used to encrypt and sign the customer SoC and firmware images, and the config engine uses the same keys to decrypt and verify these images.

4. After adjusting the settings in the Policy Editor, click the Generate button to generate a customer policy .bin file.
5. Place the customer policy .bin file in the Provision Tool directory and use the Provision Tool to program the MachXO5-NX LFMXO5-55TD device with this customer policy file.

8. Programming and Configuration

8.1. Program the BMC into CertusPro-NX Flash

Option 1: Use DediProg to program the external SPI Flash with the BMC .bit file.

Option 2: Use Lattice Radiant Programmer to program the external SPI Flash.

1. Install the following jumpers to scan the JTAG chain:
   JP12, JP19, JP51, JP52, JP53, JP54, JP55, JP56
2. Install the following jumpers to set up the CertusPro-NX device in the JTAG chain:
   JP47[1:2], JP48[3:4]
3. Install jumper JP15 to hold the MachXO5-NX LFMXO5-55TD device in reset while the CertusPro-NX device is being programmed.
4. Set up the jumpers for Flash C:
   - For Flash A: JP39[2:3], JP40[2:3], JP56[2:3]
   - For Flash B: JP39[2:3], JP40[2:3], JP57[2:3]
   - For Flash C: JP33[2:3], JP34[2:3], JP54[2:3]
   - For Flash D: JP33[2:3], JP34[2:3], JP55[2:3]

   Figure 8.1 shows the Sentry Demo Board with jumpers for BMC programming marked.

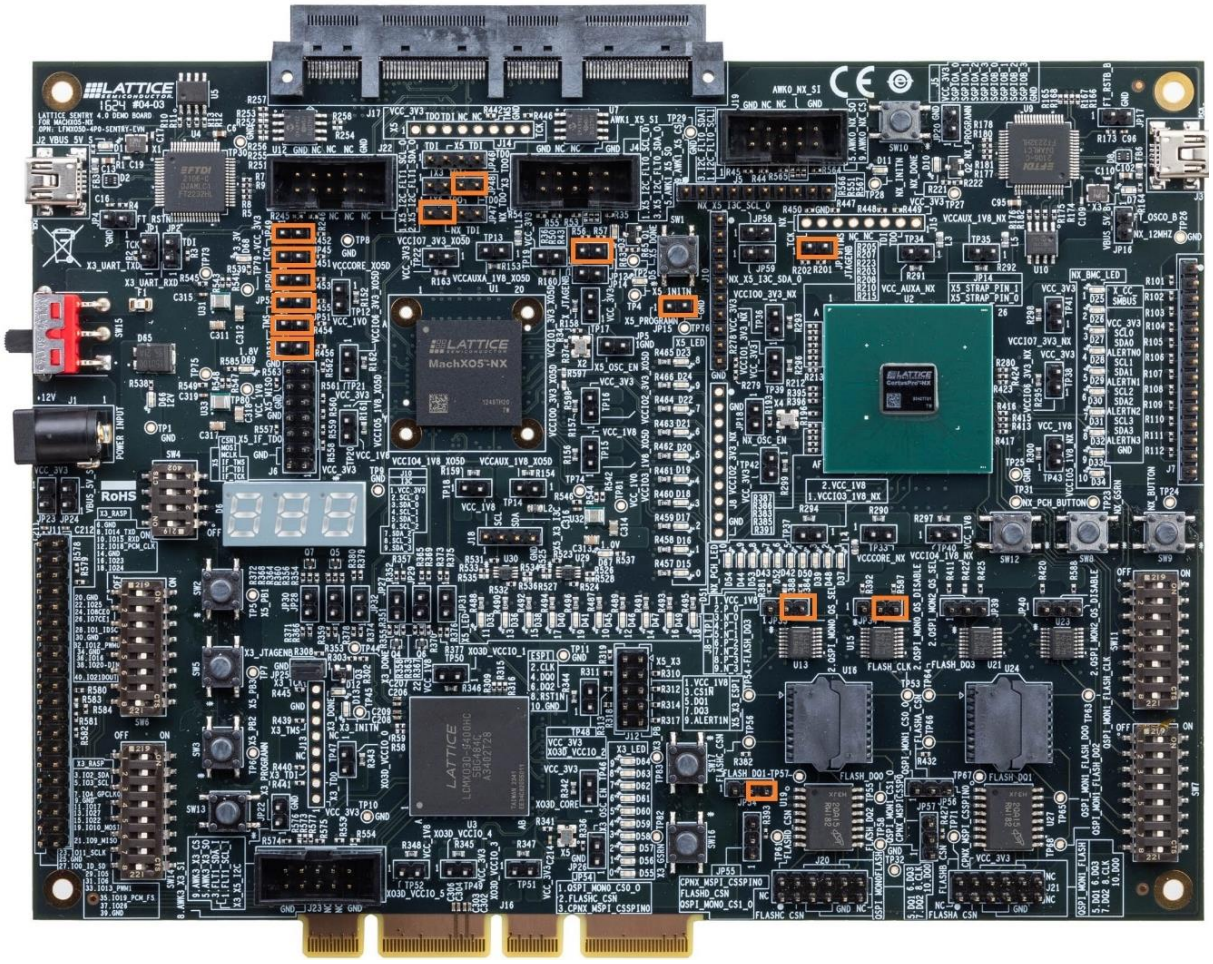

Figure 8.1. Lattice Sentry Demo Board with Jumpers for BMC Programming Marked

5. Launch Radiant Programmer.
6. Click the Scan Device icon in the ribbon to scan the device (Figure 8.2).

Figure 8.2. Lattice Radiant Programmer Scan Device Icon

7. Device Family should be LFCPNX and Device should be LFCPNX-100.
8. Double-click in the **Operation** field to open the Device Properties dialog box.
9. In Device Properties, select:
   - **Target Memory**: External SPI Flash Memory (SPI FLASH)
   - **Port Interface**: JTAG2SPI
   - **Access Mode**: Direct FLASH Programming
10. Select Erase, Program, Verify in **Operation** to open the dialog box.
11. Load the BMC_I2C.bit file to **Programming Options**: Programming file

    Note: Two OOB buses are supported, one over I2C and one over I3C. By default, the Sentry 4.0 firmware uses the I2C bus. To use the I3C OOB bus instead, program BMC_I3C.bit into the CertusPro-NX device and change the variable assignment pfr_mctp_path = MCTP_OVER_I2C to pfr_mctp_path = MCTP_OVER_I3C in the firmware.
12. Select the following SPI Flash Options (Figure 8.3):
    - Family: SPI Serial Flash
    - Vendor: Micron
    - Device: MT25QL01
    - Package: 16-Pin SOP2
13. Set SPI Programming Start address (hex) to 0x00000000.

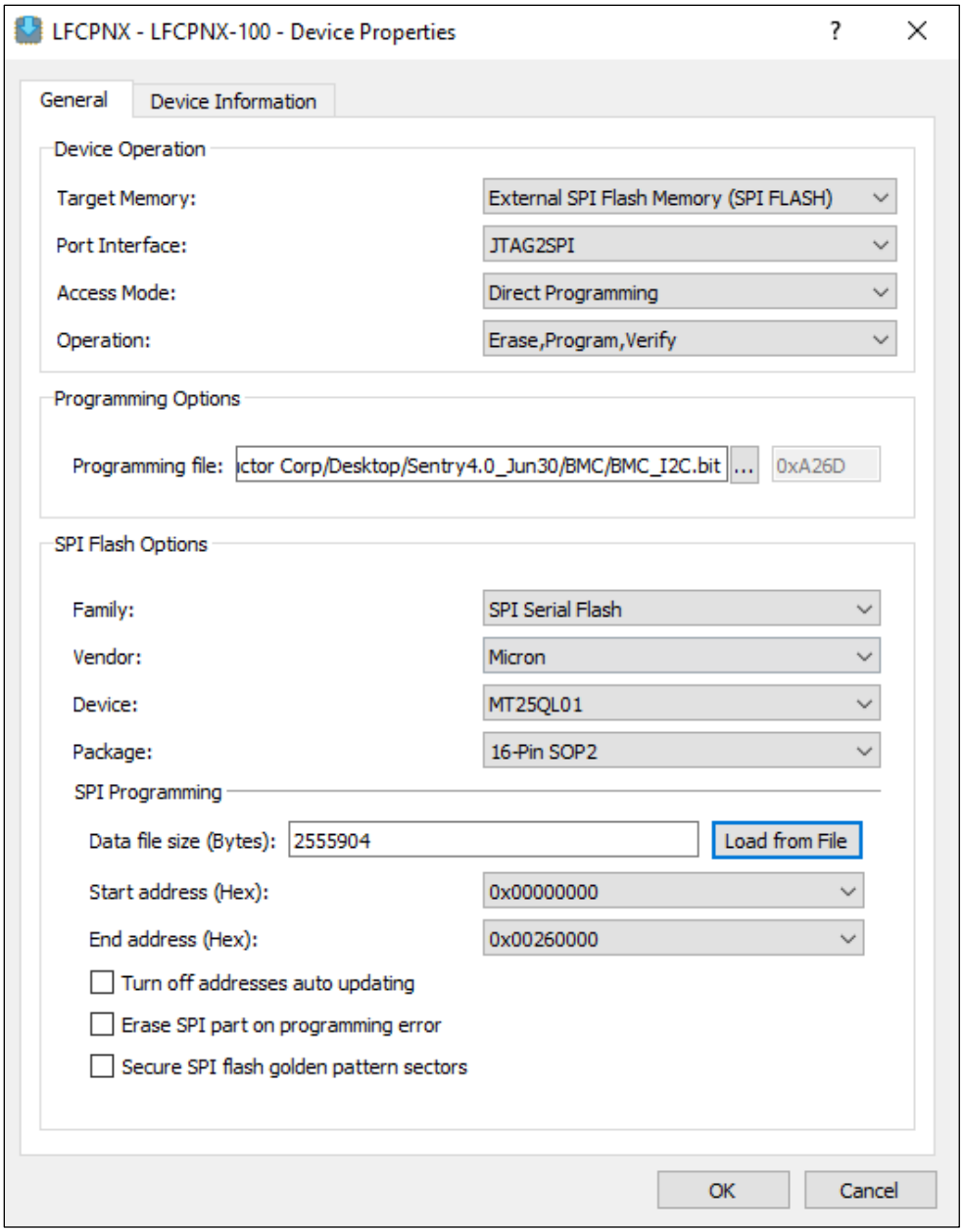

Figure 8.3. CertusPro-NX Device BMC Image Programming Options

14. Click on the Program icon to program the device and wait for a success message (Figure 8.4).

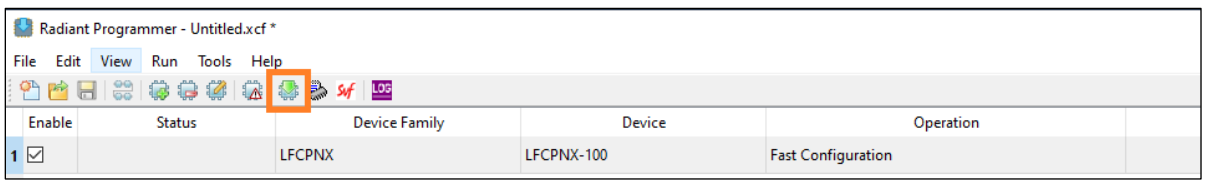

Figure 8.4. Lattice Radiant Programmer Program Device Icon

15. Remove Flash C jumpers and install JP54[1:2]. Leave JP54[1:2] installed while running the demo.

8.2. Sign the Firmware File

The .mem file generated by Lattice Propel SDK needs to be signed with an ISK private key before being programmed onto the LFMXO5-55TD device.

The ISK private key must be the same as in the ISK key pair which signed the bitstream in Lattice Radiant software, as shown in the Generate Bitstream in Lattice Radiant Software section.

The SKP tool can be used to sign the firmware file. The SKP is an internal, Lattice only tool. A customer-facing image signing tool is currently under development. Alternatively, a custom script or firmware signing tool can be used.

To sign the firmware file using the SKP tool:

1. Copy sentry40_ c.mem file to SKP_Tool_RBP folder.
2. Copy isk384.prv from the SoC project folder to SKP Tool RBP folder.
3. Sign Sentry C binary.

   Use SKP tool to sign sentry40_ c.mem file and output RiscVImage.bin.

   ```
   >> Skp.exe --genpacket --rbp_version "12345678" --ecdsaprvfile "isk384.prv" --
   ecdsaprvpwd "LATTICESEMI" --riscvimage "sentry40_ c.mem" -- keyblob
   "keyblob_normal.bin"
   ```

8.3. Program MachXO5-NX LFMXO5-55TD Device with the Provisioning Tool

The Provisioning Tool is used to program the customer SoC, Firmware image, UFM, and Customer Policy through soft JTAG.

Follow the steps below for first-time board provisioning. To reprovision the board with updated images, follow the steps in the Reprovisioning section.

1. Connect the PC to the HW-USBN-2B cable with a USB cable.
2. Make the following connections between the board and the HW-USBN-2B cable:
   - J6-1 connects to the red(VCC) wire of the HW-USBN-2B cable.
   - J6-3 connects to the white(TCK) wire of the HW-USBN-2B cable.
   - J6-5 connects to the orange(TDI) wire of the HW-USBN-2B cable.
   - J6-7 connects to the purple(TMS) wire of the HW-USBN-2B cable.
   - J6-6 connects to the brown(TDO) wire of the HW-USBN-2B cable.
   - J6-2 connects to the black(GND) wire of the HW-USBN-2B cable.
3. Set SW4 pins as follows:
   - SW4.1 = OFF
   - SW4.2 = OFF
   - SW4.3 = OFF
   - SW4.4 = don’t care
4. Power on the board.
5. The input files to the Provision Tool are shown in the following Table 8.1.

Table 8.1 Provisioning Tool Input Files

Input	Filename	Source	Description
Customer Policy	sentry_customer_policy.bin	Generated by the Lattice Propel Policy Editor tool.	Key, Policy, Lock information
Customer Image	<project_name>_impl_1.bit	Generated and signed by the Lattice Radiant software.	Signed .bit file containing SoC image
Orchestration FW Image	<project_name>.bin	Generated by Lattice Propel. Then, it is signed by the image signing tool.	Signed .bin file containing firmware image

Input	Filename	Source	Description
Customer UFM	customer_ufm.bin	Generated by the user.	Optional data or images to store in the UFM sector of the MachXO5-NX device.

Note: If any of these files have been modified, replace them in the Provision_Tool directory.

6. Run Provision Tool using .bat file:
   a. Open sentry_provision_demo.bat for editing and make sure the filenames match the files included in the Provision Tool directory.
   b. Double-click sentry_provision_demo.bat to run the provision flow.
   c. Select the correct USB port. The name of the port should have Lattice HW-USBN-2B in it, for example, Lattice HW-USBN-2B Ch A Location XXX. Input the ID of the port.
7. Run Provision Tool using the command line:
   a. Open Windows command line and navigate to the Provision Tool directory.
   b. Execute the following command:

      ```
      rot_provision_tool.exe -p {Customer Policy bin file} -x {FW Image bin file} -a {Customer Image bit file} -u {Customer UFM bin file} -m 0 -s
      ```

   c. Select the correct USB port. The name of the port should have Lattice HW-USBN-2B in it, for example, Lattice HW-USBN-2B Ch A Location XXX. Input the ID of the port.
8. After provisioning is finished, the system triggers a soft reboot.

8.4. Reprovisioning

After provisioning the board for the first time, subsequent updates are made by reprovisioning.

1. Connect the Lattice 2B Programming cable to the board and the PC as described in the Program MachXO5-NX LFMXO5-55TD Device with the Provisioning Tool section, Steps 1-2.
2. Set SW4 pins as follows:
   - SW4.1 = OFF
   - SW4.2 = ON
   - SW4.3 = OFF
   - SW4.4 = don't care
3. Power on the board.
4. Copy over any of the four Provision Tool input files from Table 8.1 into the Provision Tool directory. Note that none of these files are required. Only the files in need of updating should be included.
5. Run Provision Tool using the command line:
   a. Open the Windows command line and navigate to the Provision Tool directory.
   b. Execute the following command, with optional arguments (Figure 8.5):

      ```
      rot_provision_tool.exe -p {Customer Policy bin file} -x {FW Image bin file} -a {Customer Image bit file} -u {Customer UFM bin file}
      ```

   c. Select the correct USB port. The name of the port should have Lattice HW-USBN-2B in it, for example, Lattice HW-USBN-2B Ch A Location XXX. Input the ID of the port.

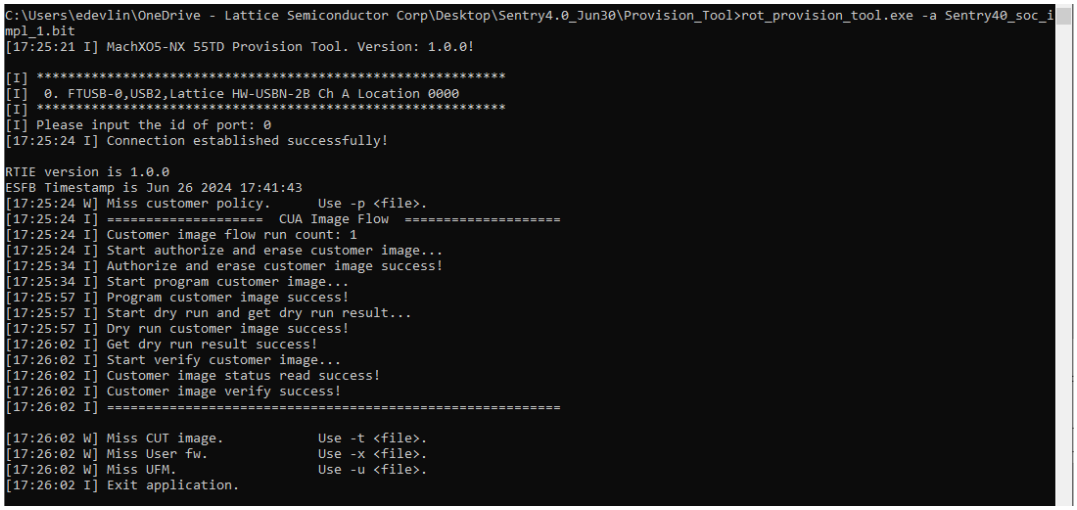

Figure 8.5. Sentry Provision Tool Reprovision CUA Image

d. After provisioning finishes successfully, execute the following command to set the Provision Done bit (Figure 8.6):

```
rot_provision_tool.exe -s
```

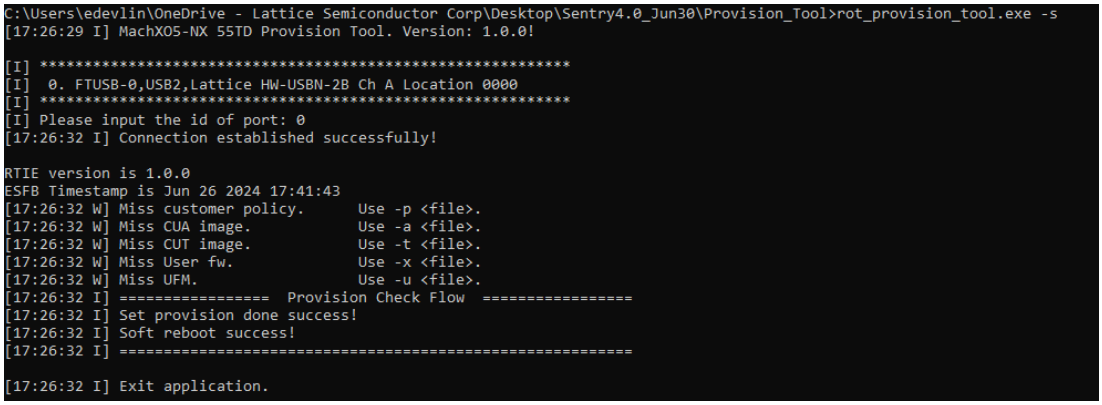

Figure 8.6. Provision Tool Set Provision Done

6. Power cycle the board to boot from the updated image.

9. Running the Demo and Using Demo Tools

9.1. Basic Demo (UART and LEDs)

After programming the BMC image into CertusPro-NX Flash C and running the Provision Tool to program the customer image, firmware image, customer policy, and optional UFM into MachXO5-NX LFMXO5-55TD device, the board is ready to run the demo.

1. Connect a USB cable between J2 and a PC for UART output from MachXO5-NX LFMXO5-55TD. Make sure JP1 and JP2 are not installed.
2. Make sure SW4 is set as follows:
   - SW4.1 = OFF
   - SW4.2 = OFF
   - SW4.3 = OFF
   - SW4.4 = don't care
3. Power on the board.
4. Open a serial terminal emulator, such as PuTTY.
5. Connect to the higher number of the two COM ports. The baud rate is 115200 (Figure 9.1).

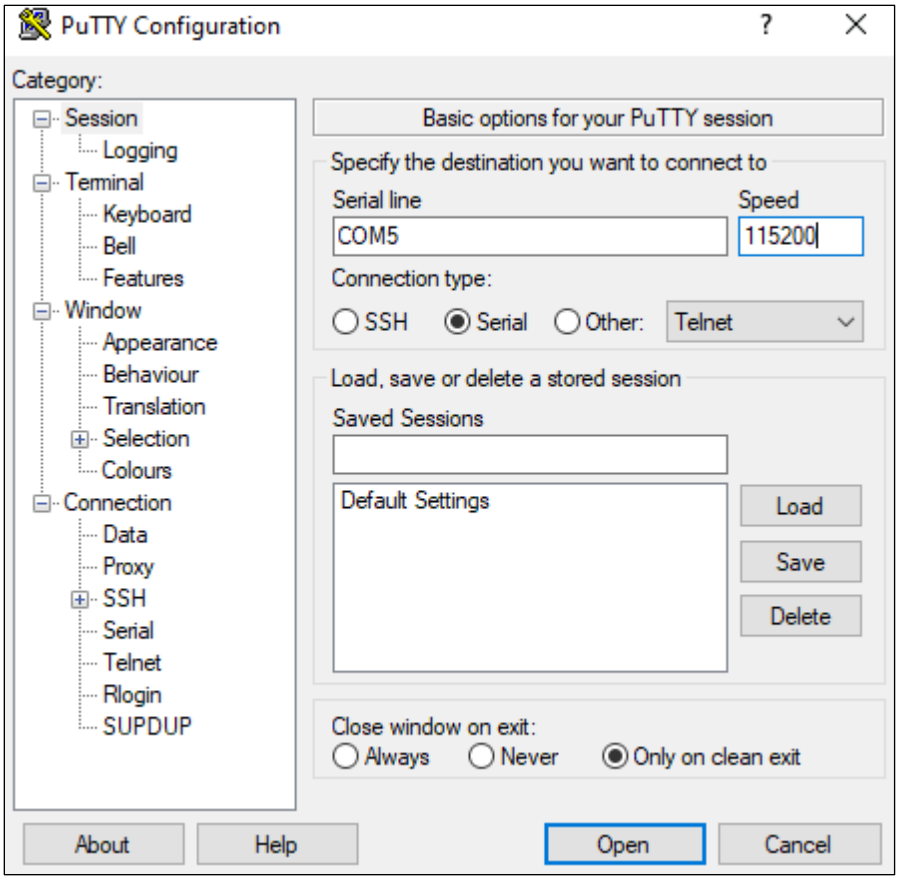

Figure 9.1. PuTTY Configuration

6. UART print statements should be visible on the terminal (Figure 9.2).

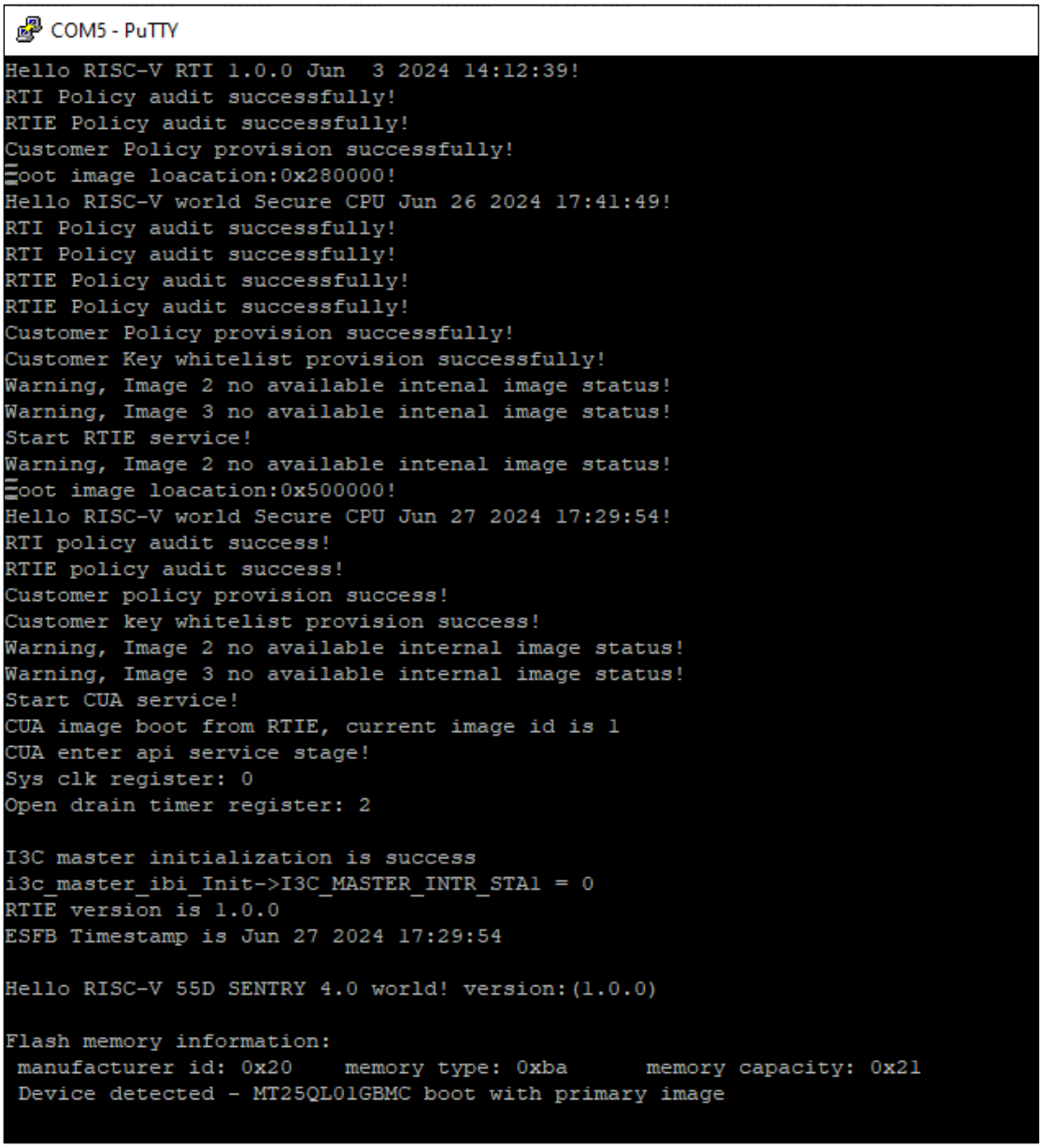

Figure 9.2. MachXO5-NX LFMXO5-55TD Device UART Output

7. The 7-segment LED should display *4*.
8. After the BMC boots, the following LEDs should be on:
   - D53, D43, D52, D42, D50, D39, D48, D37
   - D25, D26, D27, D28, D29, D30, D31, D32

9.2. Demo GUI

There are two demo GUIs included in the Sentry 4.0 patch. Both can be accessed from Lattice Propel SDK, through the Lattice Tools menu. The Lattice Sentry Demo GUI in the Lattice Sentry Tools for MachXO5-NX LFMXO5-55TD device demonstrates PFR functionality. The Demo GUI in the Lattice RoT Tools for MachXO5-NX LFMXO5-55TD device demonstrates Root of Trust functionality.

The Lattice Sentry Demo GUI connects via UART to the CertusPro-NX device, which represents a BMC in a server environment. The user selects commands from a menu of options in the GUI. When the user clicks **Send Command** in

the demo GUI, the CertusPro-NX (BMC) device sends a read or write command through the OOB channel to the LFMXO5-55TD device. As the LFMXO5-55TD device executes the command, there may be print statements on LFMXO5-55TD's UART terminal. After the command is executed, a success or failure message is sent from the LFMXO5-55TD device to the CertusPro-NX device, and a message is printed on the Demo GUI console.

Before running the Lattice Sentry demo GUI, make sure all DEBUG macros in the firmware are defined as 1 instead of 0. This is to ensure verbose UART output from the LFMXO5-55TD Sentry design as it responds to commands from the BMC through the demo GUI. These macros are located in the firmware file pfr_conf.h (Figure 9.3).

```
89  #define MAIN_DEBUG              1
90  #define OOB_DEBUG                  1
91  #define MANIFEST_DEBUG           1
92  #define FLASH_MONITOR_DEBUG     1
93  #define LOG_DEBUG                  1
94  #define FLASH_COMMON_DEBUG         1
95  #define FLASH_ID_DEBUG             1
96  #define MCTP_DEBUG                 1
97  #define SMBUS_MCTP_DEBUG              1
```

Figure 9.3. DEBUG macros in pfr_conf.h

To run the Lattice Sentry Demo GUI:

1. Connect a USB cable from J2 to your PC and open a serial terminal for the higher number port with baud rate 115200 to connect to the LFMXO5-55TD device.
2. Connect a second USB cable from J3 to your PC. Do not open a serial terminal for this connection.
3. Launch Lattice Propel SDK and open the Sentry 4.0 firmware workspace.
4. From the menu, select **Lattice Tools** -> **Lattice Sentry Tools for MachXO5-55TD** -> **Lattice Sentry Demo GUI** (Figure 9.4).

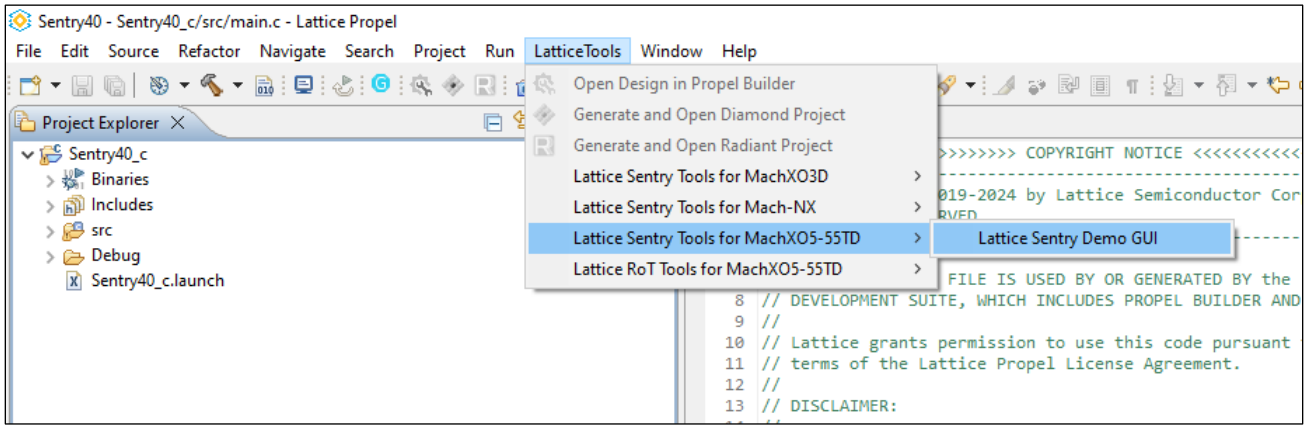

Figure 9.4. Launch Lattice Sentry Demo GUI

5. When the Demo GUI window opens, click **Scan Ports** (Figure 9.5).

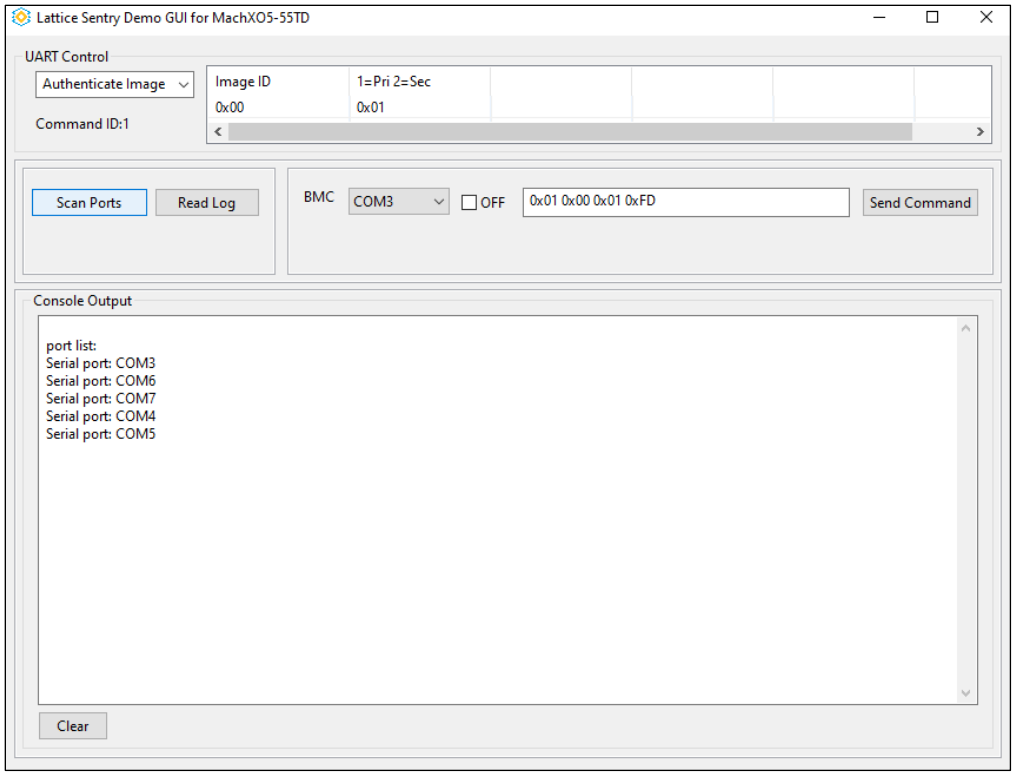

Figure 9.5. Lattice Sentry Demo GUI Scan Ports

6. Click the drop-down menu next to **BMC** and select the lower numbered port for the USB cable connected to J3 (Figure 9.6).

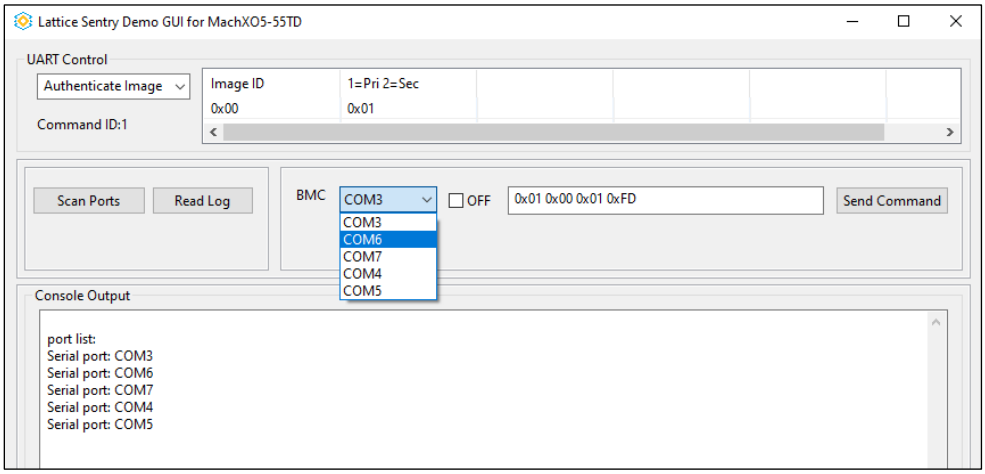

Figure 9.6. Lattice Sentry Demo GUI Select BMC Port

7. Click the checkbox next to **OFF** to connect the BMC to the selected port.
8. Make sure a success message is displayed on the Demo GUI console (Figure 9.7).

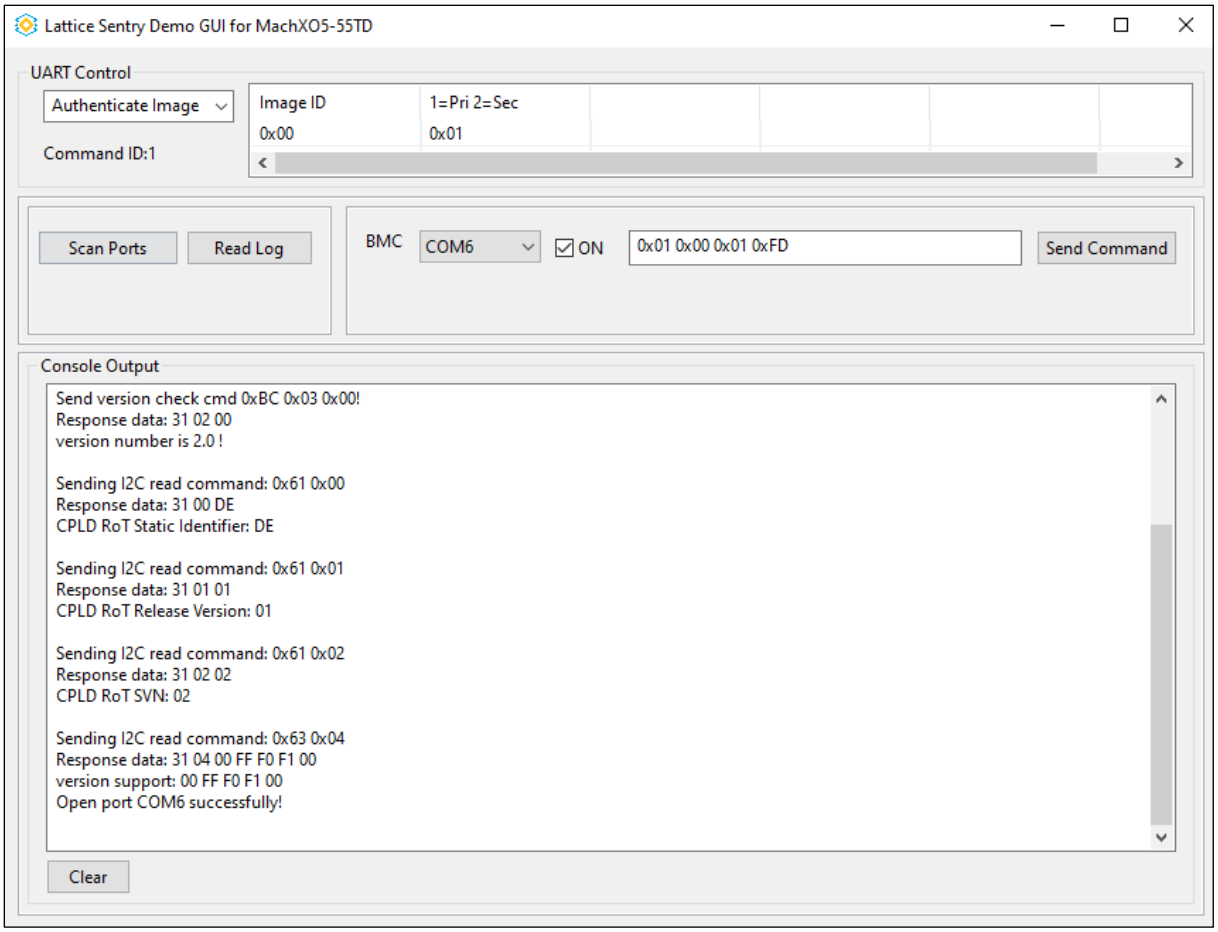

Figure 9.7. Lattice Sentry Demo GUI BMC Connection Success

9. Select an OOB command from the drop-down menu under UART control. Modify the command options as desired.
10. Click **Send Command** to prompt the BMC to send a UART command to the LFMXO5-55TD device. In this example, the Authenticate Image command has been sent, with the options of Image ID 0 and Primary Image (0x01).
11. After the command is sent, data should be displayed on the LFMXO5-55TD serial terminal from UART print statements. A **Done/Success** message should be displayed on the demo GUI terminal (Figure 9.8).

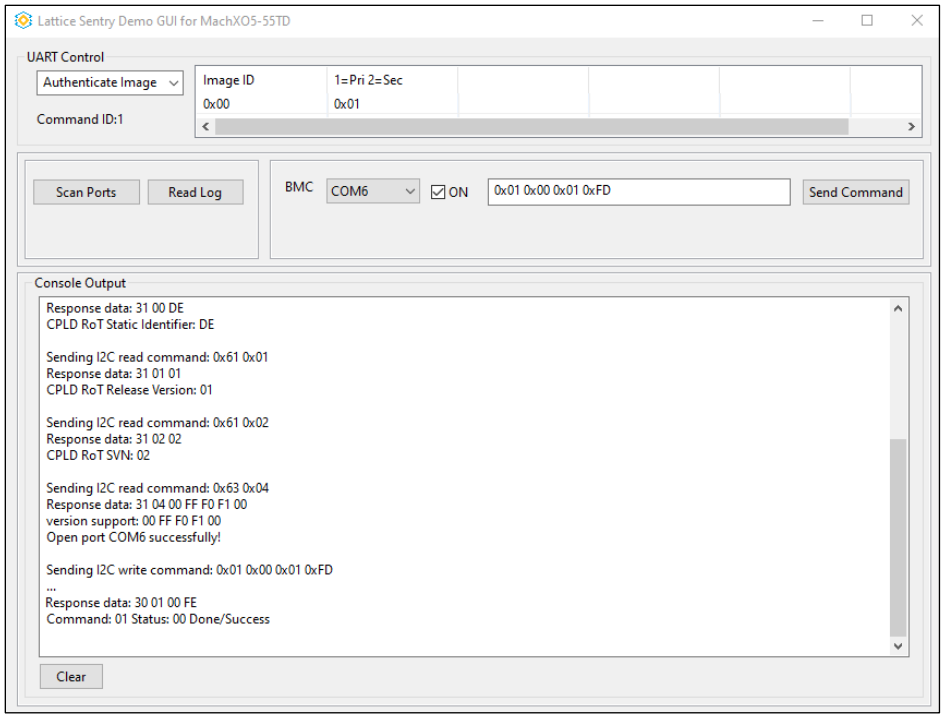

Figure 9.8. Lattice Sentry Demo GUI Send Command

12. Click **Read Log** to view the most recent log entry. Continue clicking Read Log to view earlier log entries (Figure 9.9).

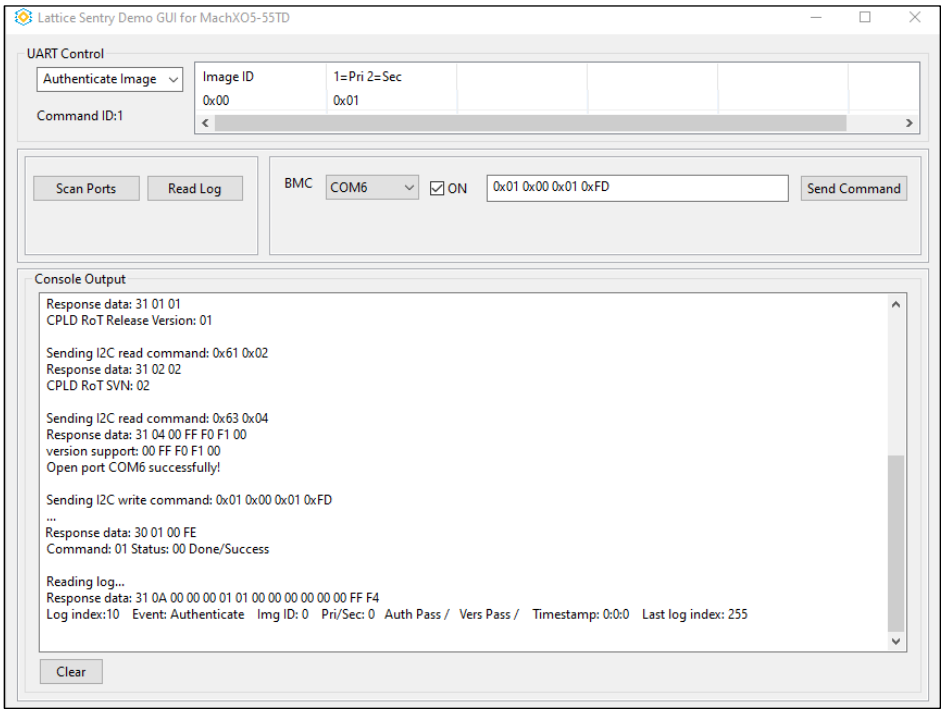

Figure 9.9. Lattice Sentry Demo GUI Read Log

13. Some Demo GUI commands require Secure OOB mode. To enter Secure OOB mode, select **Enable Secure Session** from the UART Control drop-down menu (Figure 9.10).

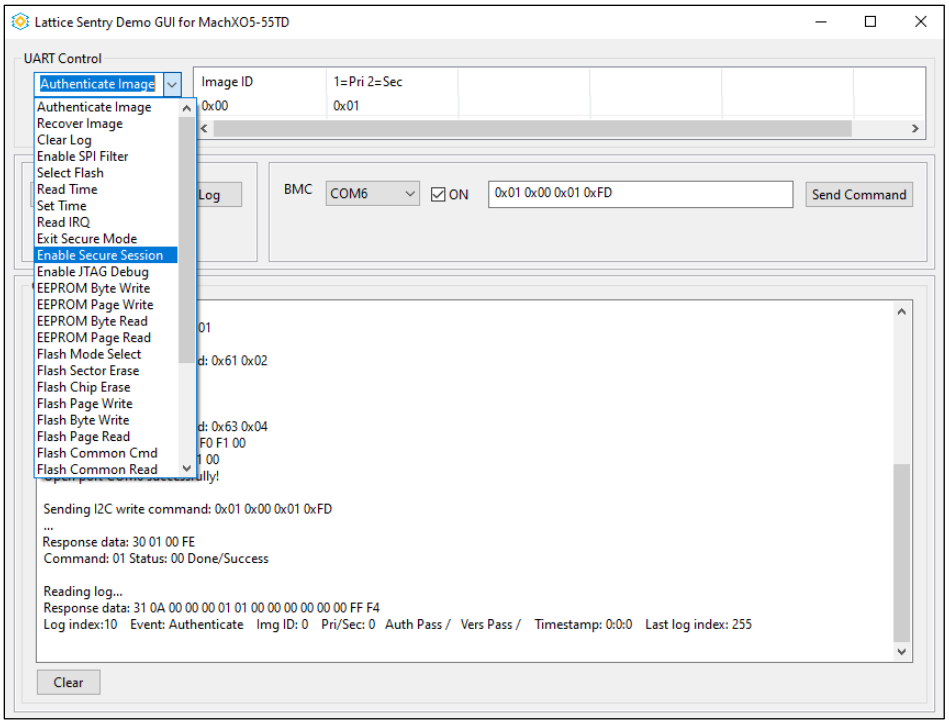

Figure 9.10. Lattice Sentry Demo GUI Enable Secure Session

14. Change the input argument to 0x02 to use ECDSA-384 authentication (Figure 9.11).

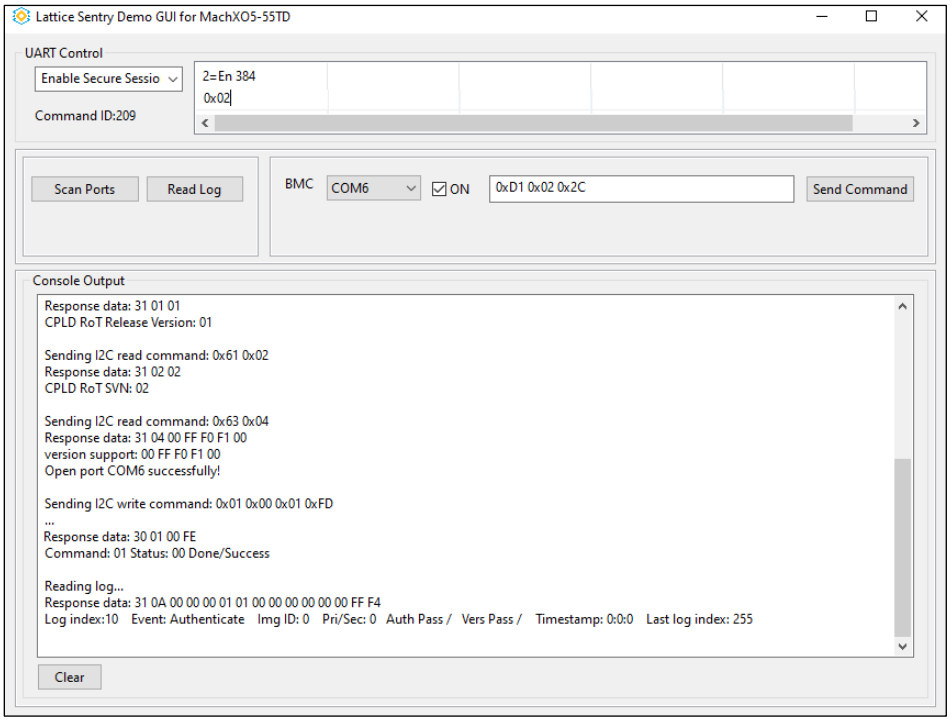

Figure 9.11. Enable Secure Session Options

15. Click **Send Command**.
16. Wait for a **Done/Success** message to appear on the Demo GUI console. Now Secure OOB mode is enabled and all commands can be run (Figure 9.12).

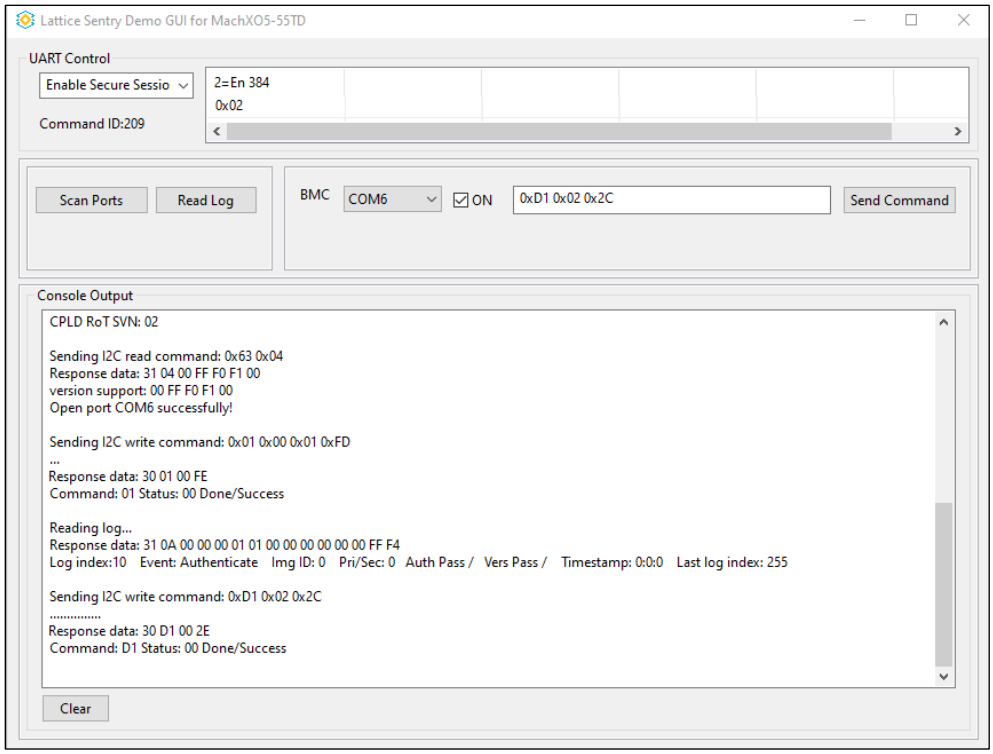

Figure 9.12. Lattice Sentry Demo GUI Secure OOB Mode Enabled

10. Troubleshooting

10.1. Verification Error in Lattice Radiant Programmer

Problem Description: The following error message shows during programing devices in Lattice Radiant Programmer.

```
ERROR - Verification Error...when Processing function: 'CHECK_ID'
```

Solution: Select **Use custom Clock Divider** and change the TCK Divider Setting to 5.

10.2. SoC Project Does Not Fully Generate

Problem Description: Lattice Propel Builder generates a blank project, or a project with components missing.

Solution: Make sure the correct version of all IPs are installed in Lattice Propel Builder before generating the project.

References

- MachXO5-NX Family Devices web page
- Lattice Sentry Solution web page
- Lattice Radiant FPGA design software
- Lattice Propel Design Environment web page
- Lattice Radiant Software User Guide
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- 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, please refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.0, July 2024

Section	Change Summary
All	Production release.

www.latticesemi.com