

Technical Note

# NXP i.MX 8X: Booting via debug interface

May 2023



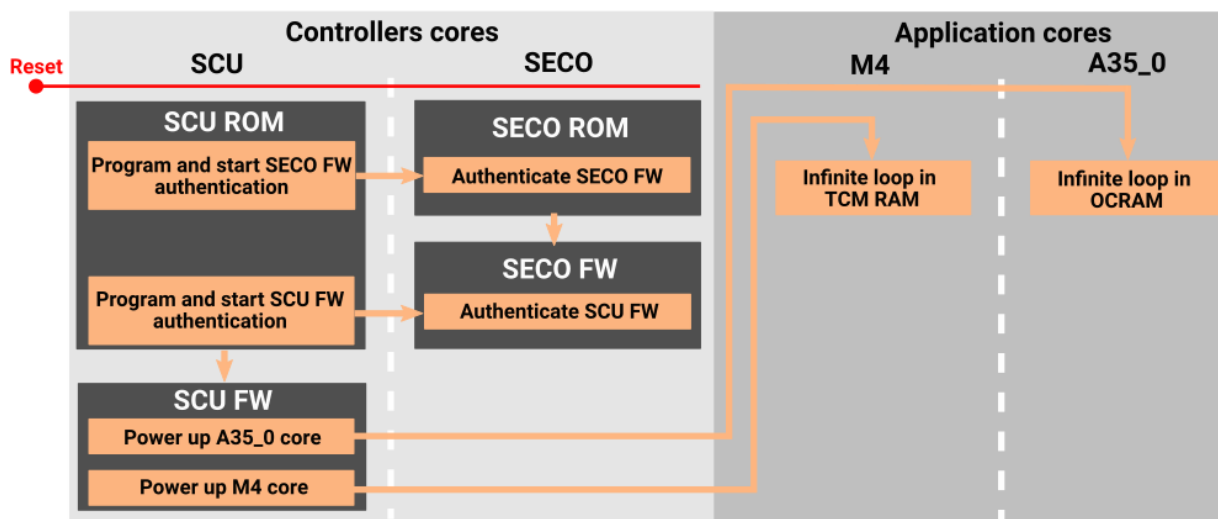
A **TASKING** Company

This Technical Note describes booting the NXP i.MX 8X devices via debug interface using winIDEA to enable programming and debugging of the application cores.

**Tool requirements:**

- ✓ winIDEA 9.17.130 or newer
- ✓ BlueBox iC5700, iC5000
- ✓ ARM HSSTP II Active Probe or CoreSight Debug Adapter

Beside application cores NXP i.MX 8X devices feature two dedicated controller cores: System controller (SCU) and Security controller (SECO). After reset only controller cores are started while the application cores remain in reset. Proper powering and configuring of both controller cores will initialize booting of the application cores.



Both controller cores ROMs and firmwares facilitate the JTAG booting of the application cores. While the internal ROM code (typically) cannot be altered, the firmware can be adjusted to the target board requirements.

### SCU and SECO firmware

Both SCU and SECO firmwares consist of two separate parts: Firmware header and Firmware image. Both parts are essential for the successful boot. The firmware images for the boot procedure cannot be uploaded from a typical boot medium (eMMC flash, QSPI flash, SD card), but need to be provided by a development tool. SCU and SECO firmware images and headers compatible and tested with NXP's MCIMX8QXP-CPU development kit are distributed with winIDEA IDE and are loaded by the default Connect to SoC procedure for establishing debug connection with the NXP i.MX 8X processor family devices.

## winIDEA Configuration

The default initialization script `$(SFR_FILE_DIR)\iMX8_ConnectToSoC.cpp (g_device=iMX8X)` loads the default firmware.



Following error messages when establishing a debug session:

- Failed to boot SCU
- Failed to boot SECO
- Failed to power up application cores

indicate that the default firmware is not compatible with the connected target device. To specify custom SECO / SCU firmware, parameters must be provided to the initialization script:

```
g_iMX8X_SECOFW_hdr=path_to_header, g_iMX8X_SECOFW_img=path_to_image,
g_iMX8X_SCUFW_hdr=path_to_header, g_iMX8X_SCUFW_img=path_to_image
```

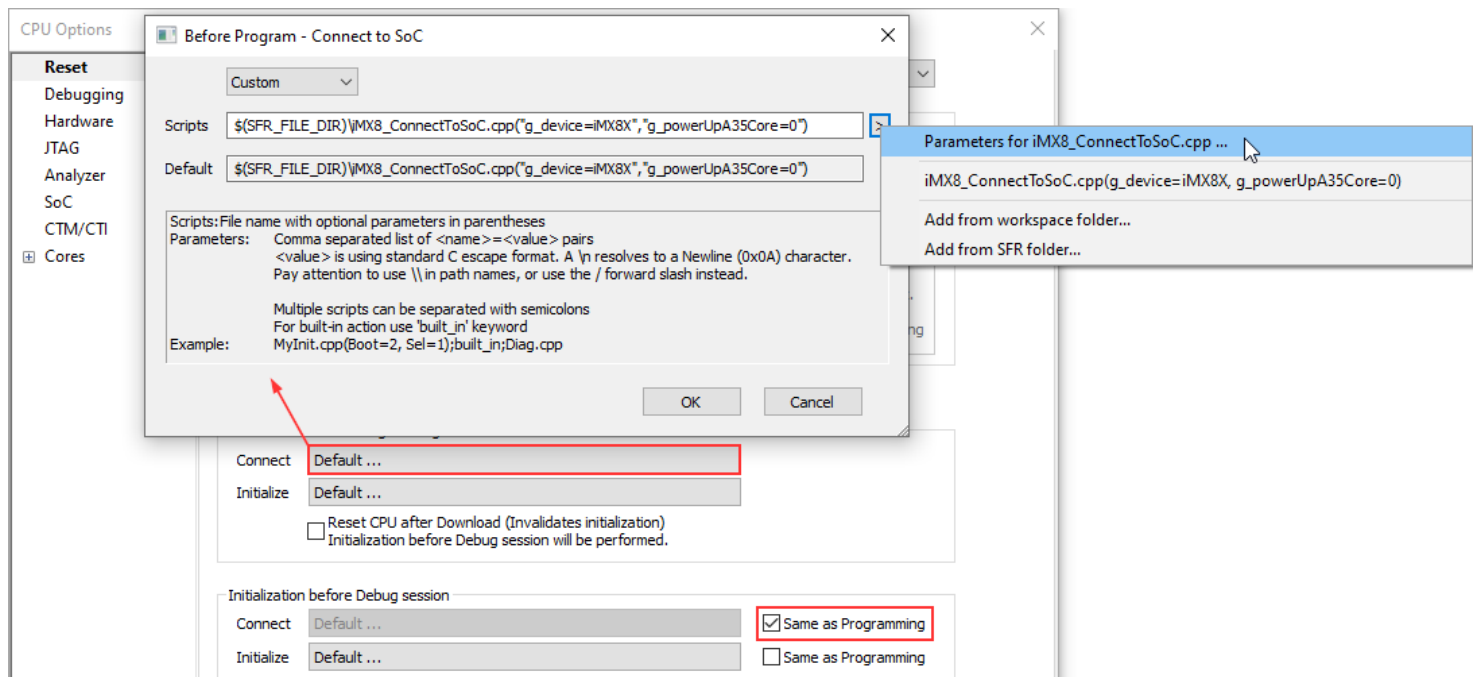
Follow the described configuration procedure to use a custom firmware:

1. Open *Hardware menu / CPU Options / Reset tab*.
2. Press *Default Initialization before Programming* section.
3. Select *Custom* in the drop-down menu.
4. Press the arrow  button and add the script *iMX8\_ConnectToSoc.cpp*.
5. Press the arrow  button and select Parameters.
6. Specify the path of your custom firmware header and image for SECO and SCU as the parameters of the script and confirm with OK.

Example script parameters:

```
$(SFR_FILE_DIR)\iMX8_ConnectToSoc.cpp(g_iMX8X_SECOFW_img=C:/SECO_image.bin,
g_iMX8X_SECOFW_hdr=C:/SECO_header.bin,g_iMX8X_SCUFW_hdr=C:/SCU_header.bin,
g_iMX8X_SCUFW_img=C:/SCU_image.bin)
```

7. Check option *Same as Programming* in *Initialization before Debug session / Connect* and confirm with OK.



iSYSTEM doesn't provide custom firmware. Consult with your silicon vendor representative.