IMXWQSG

i.MX Windows IoT Quick Start Guide

Rev. 1.5.2 — 11 November 2024 **User guide**

Document information

Information	Content
Keywords	i.MX, Windows IoT
Abstract	This document guides you through the process of downloading and running this release package. It only explains how to download and run the default release image with the default configuration. For details on using the release package, see the Windows IoT User's Guide (IMXWUG) included in this release package.

1 Overview

This document guides you through the process of downloading and running this release package. It explains how to download and run the default release image with the default configuration. For details on using the release package, see the Windows IoT User's Guide (IMXWUG) included in this release package.

2 Hardware requirements

The hardware requirements for using this release package are as follows:

Supported evaluation boards (for more information on supported board revisions, see Release Notes):

- i.MX 8M Mini EVK
- i.MX 8M Nano EVK
- i.MX 8M Plus EVK
- i.MX 8M Quad EVK
- i.MX 8QuadXPlus MEK (Silicon Revision C0)
- i.MX 93 EVK (Silicon Revision A1)
  - **NOTE**: i.MX 93 EVK Silicon Revision A0 is no longer supported !!!

Additional equipment:

- microSD card (minimum 8 GB)
- i.MX Mini SAS cable with IMX-MIPI-HDMI adapter (MIPI-DSI to HDMI converter) or IMX-LVDS-HDMI adapter (LVDS to HDMI converter), depending on the evaluation board/SoC capability. For more information, see the specific board chapter.

3 Software requirements

- Binary drivers and firmware - either downloaded from nxp.com or built locally.
- Windows IoT operating system.
  - Note: Windows 11 IoT Enterprise is only compatible with the NXP i.MX 93. It is NOT compatible with the NXP i.MX 8M family or the NXP i.MX 8QuadXPlus
  - If you are evaluating Windows 10 IoT Enterprise (for example, NXP i.MX 8M family or NXP i.MX 8QuadXPlus):
    - Visual Studio Subscription portal my.visualstudio.com
      - At the portal, click *Downloads* -> *Windows 10* -> Search for “Windows 10 IoT Enterprise LTSC 2021” or “Windows 10 IoT Enterprise 2021”.
      - The default architecture is set to x64, click the dropdown menu to change it to Arm64 and download the DVD
      - The recommended DVD is:
      - en-us_windows_10_iot_enterprise_ltsc_2021_arm64_dvd_e8d4fc46.iso
      - SHA256: D265DF49B30A1477D010C79185A7BC88591A1BE4B3EB690C994BED828EA17C000
  - If you are evaluating Windows 11 IoT Enterprise (for example, NXP i.MX 93):
    - Download the free 90-day evaluation using the following link
      - After filling out the registration page, select the Arm64 edition download and download the DVD ISO:
        - 26100.1.240331-1435.ge_release_CLIENT_IOT_LTSC_EVAL_A64FRE_en-us.iso
  - For both Windows 10 IoT Enterprise or Windows 11 IoT Enterprise, you can also get the OS through microsoftoem.com facilitated by a Windows IoT OS distributor
    - To find a distributor, visit Windows IoT Distributor Information
- Windows ADK for Windows 10 and Windows PE add-on for ADK, version 2004.

3.1 Serial logging setup

To help troubleshoot issues during boot, use the USB micro-B port on i.MX EVK boards to output U-Boot and the UEFI firmware serial debug logs to a host PC. The USB micro-B port on the EVK presents a virtual serial port to the host PC that can be read by the common Windows serial terminal applications such as HyperTerminal Tera Term, or PuTTY.

1. Connect the target and the PC using the cable mentioned above.
2. Open the Device Manager on the PC, locate the Enhanced Virtual serial device and note the COM port number.
3. Open the terminal on the PC. Configure the Enhanced Virtual serial/COM port to 921600 baud/s, 8-bit, one-stop bit.

For the Host PC to recognize the i.MX device’s virtual serial port, you might need to download and install drivers for the i.MX’s USB to serial converter.

The i.MX 8M EVK uses the CP210x USB to serial bridge. The CP210x driver can be found here.

The i.MX 8M Mini EVK uses the FT2232D USB to serial bridge. The FT2232D driver can be found here.

Note: The order in which serial/COM ports appear in Windows can differ depending on the board and Windows host.

3.2 Windows driver debugging over serial cable

According to the following information the target board debug port can be configured by:

bcdedit /dbgsettings serial debugport:n baudrate:rate

In this BSPs, the serial port number and baud rate are not influenced by this command and must be configured as described below.

Target board serial port used for debugging is defined in the ACPI Dbg2.aslc (\win10-iot-bsp\mu_platform_nxp \Silicon\ARM\NXP\iMX8Pkg\AcpiTables\Dbg2.aslc)and can be changed in the win10-iot-bsp\mu_platform_nxp \NXP\MX8M_PLUS_EVK\MX8M_PLUS_EVK.dsc file, parameter PcdKdUartInstance.

It is supposed that this serial port is fully initialized by U-Boot. The baud rate can be changed in the U-Boot configuration file (for example, win10-iot-bsp\uboot-imx\configs\imx8mq_evk_nt_uuu_defconfig, parameter CONFIG_BAUDRATE)BSP must be rebuild after changes in the source code. Use buildme64.sh -b <BOARD_NAME> -t all -c to be sure that all changes are applied.

The default debug port baud rate is set in U-Boot is 921600. Baud rates higher than 115200 may not work correctly with newer versions of the Host operating system.

3.3 References

For more information about Windows IoT Enterprise, see Microsoft online documentation.

The following quick start guides available on the NXP website contain basic information on the board and setting it up:

- i.MX 8M Quad Evaluation Kit Quick Start Guide
- i.MX 8M Mini Evaluation Kit Quick Start Guide
- i.MX 8M Nano Evaluation Kit Quick Start Guide
- i.MX 8M Plus Evaluation Kit Quick Start Guide
- 8QuadXPlus Multisensory Enablement Kit Quick Start Guide
- i.MX 93 Evaluation Kit Quick Start Guide

4 Working with the i.MX 8M Mini EVK Board

4.1 Board hardware

The figures below show different components of the i.MX 8M Mini EVK LPDDR4 board.

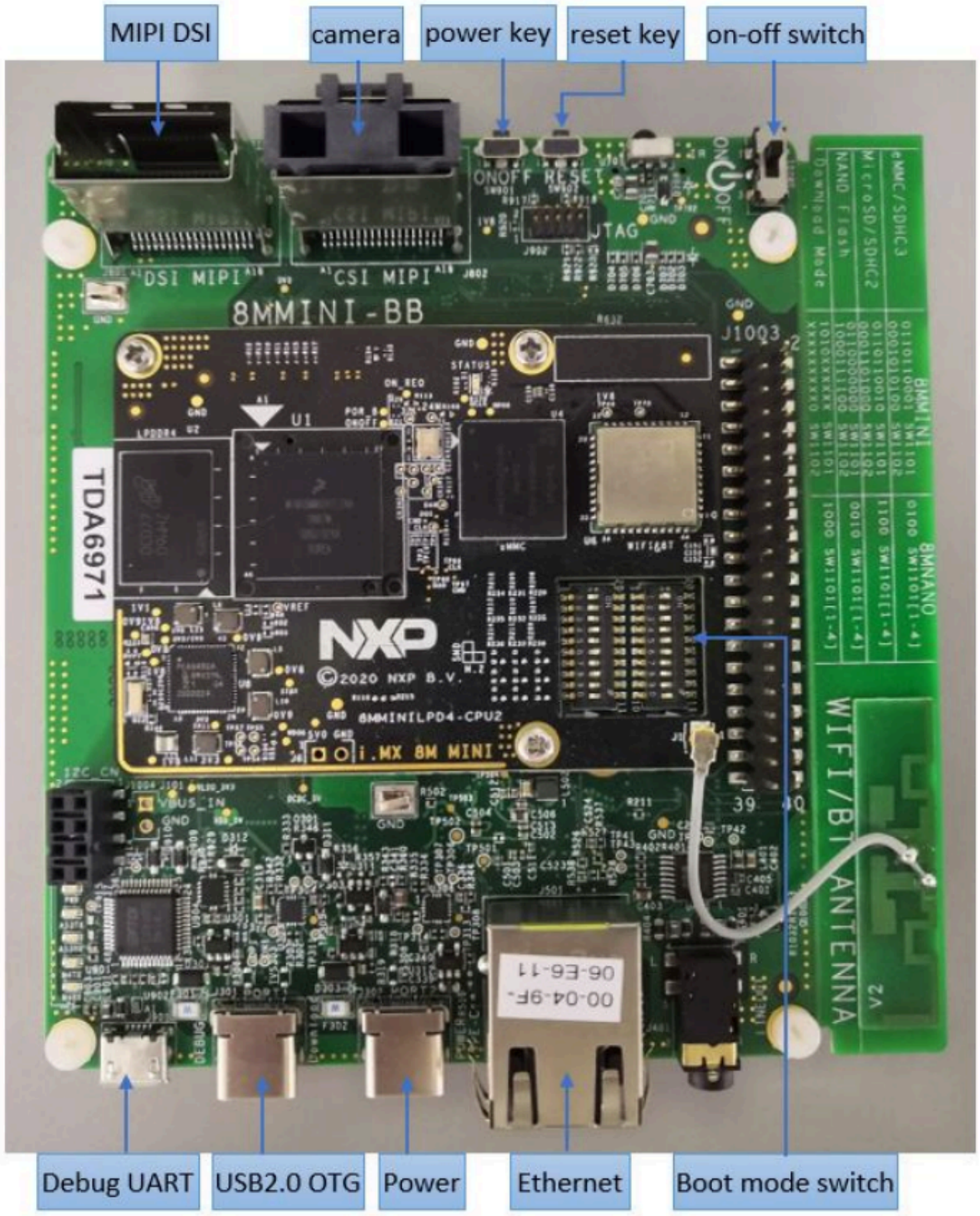


Figure 1. i.MX 8M Mini EVK LPDDR4 board

Figure 2. i.MX Mini SAS cable with DSI-to-HDMI adapter IMX-MIPI-HDMI

Figure 3. OV5640 MIPI CSI-2 camera

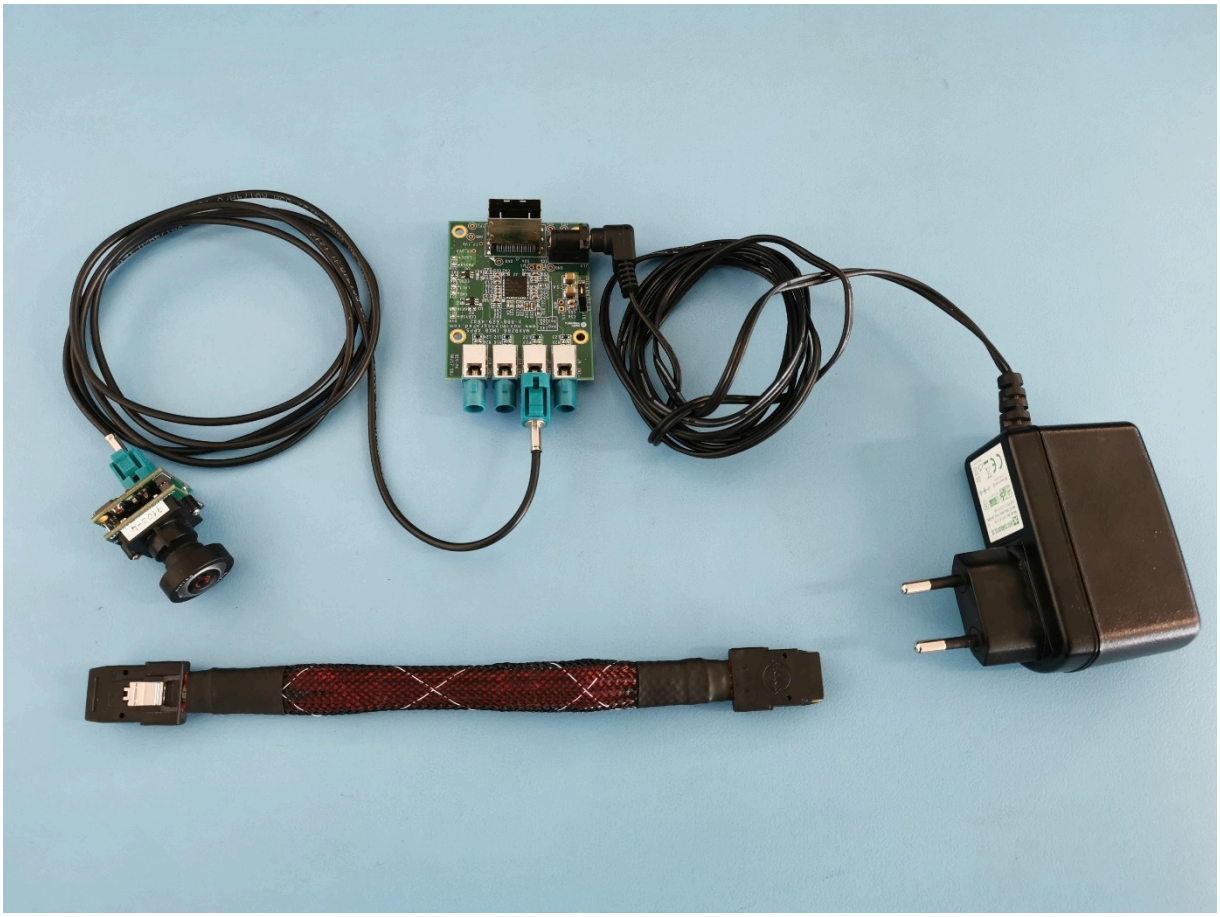


Figure 4. OV10635 Camera set

4.2 Display interface support

The MIPI-DSI to HDMI converter IMX-MIPI-HDMI is supported. The converter is connected to the MIPI-DSI display interface on the EVK board (see Figure 1) with a mini-SAS cable (see Figure 2). Use a standard HDMI cable to connect an HDMI compatible display.

5 Working with the i.MX 8M Nano EVK Board

5.1 Board hardware

The figures below show different components of the i.MX 8M Nano EVK board.

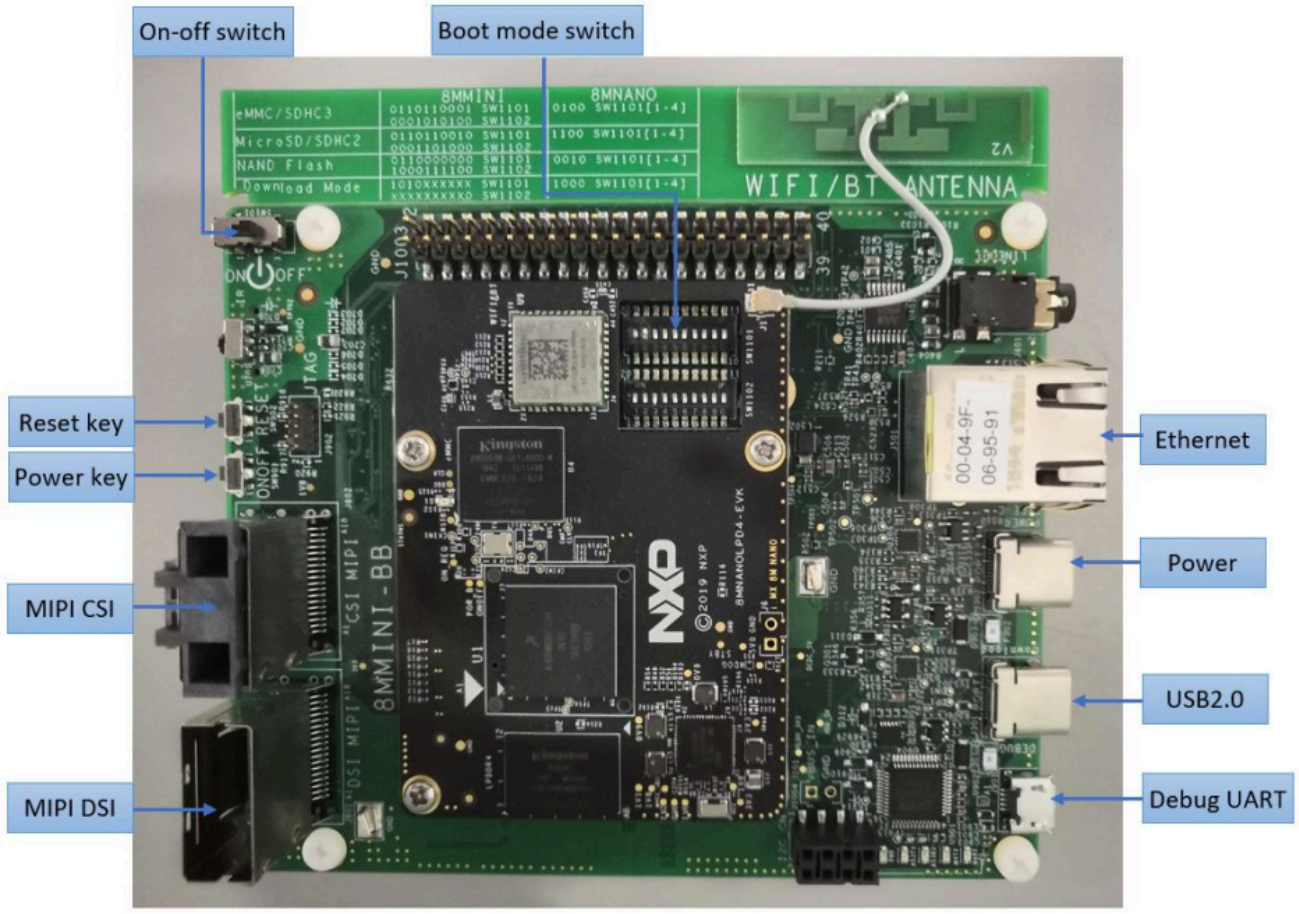


Figure 5. i.MX 8M Nano EVK board

Figure 6. i.MX Mini SAS cable with DSI-to-HDMI adapter IMX-MIPI-HDMI

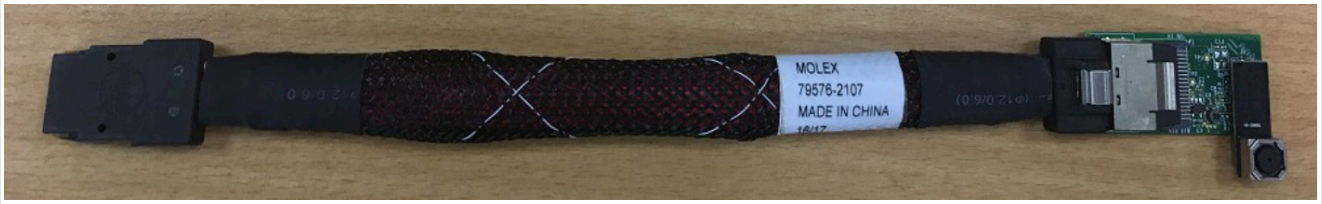


Figure 7. OV5640 MIPI CSI-2 camera

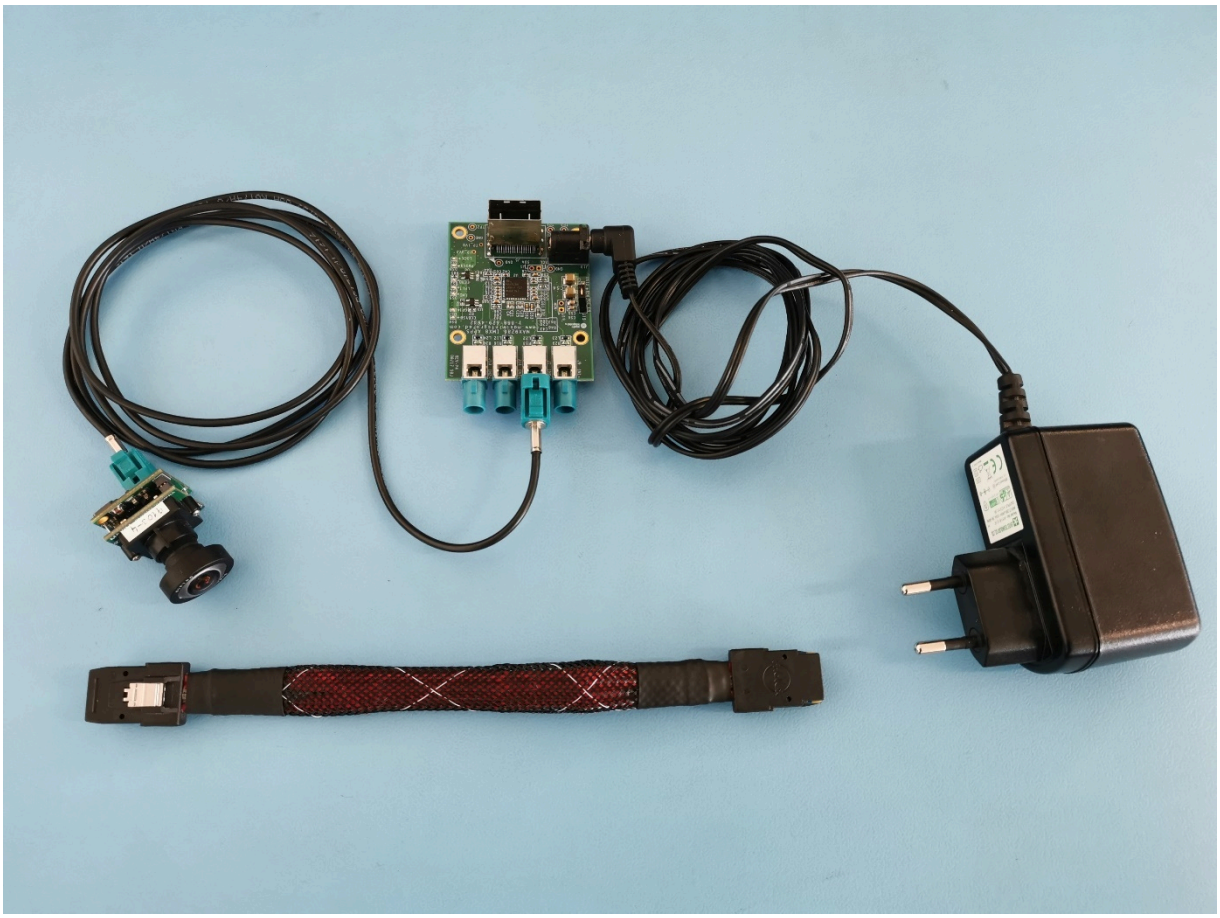


Figure 8. OV10635 Camera set

5.2 Display interface support

The MIPI-DSI to HDMI converter IMX-MIPI-HDMI is supported. The converter is connected to the MIPI-DSI display interface on the EVK board (see Figure 5) with a mini-SAS cable (see Figure 6). Use a standard HDMI cable to connect an HDMI compatible display.

6 Working with the i.MX 8M Plus EVK Board

6.1 Board hardware

The figures below show different components of the i.MX 8M Plus EVK board.

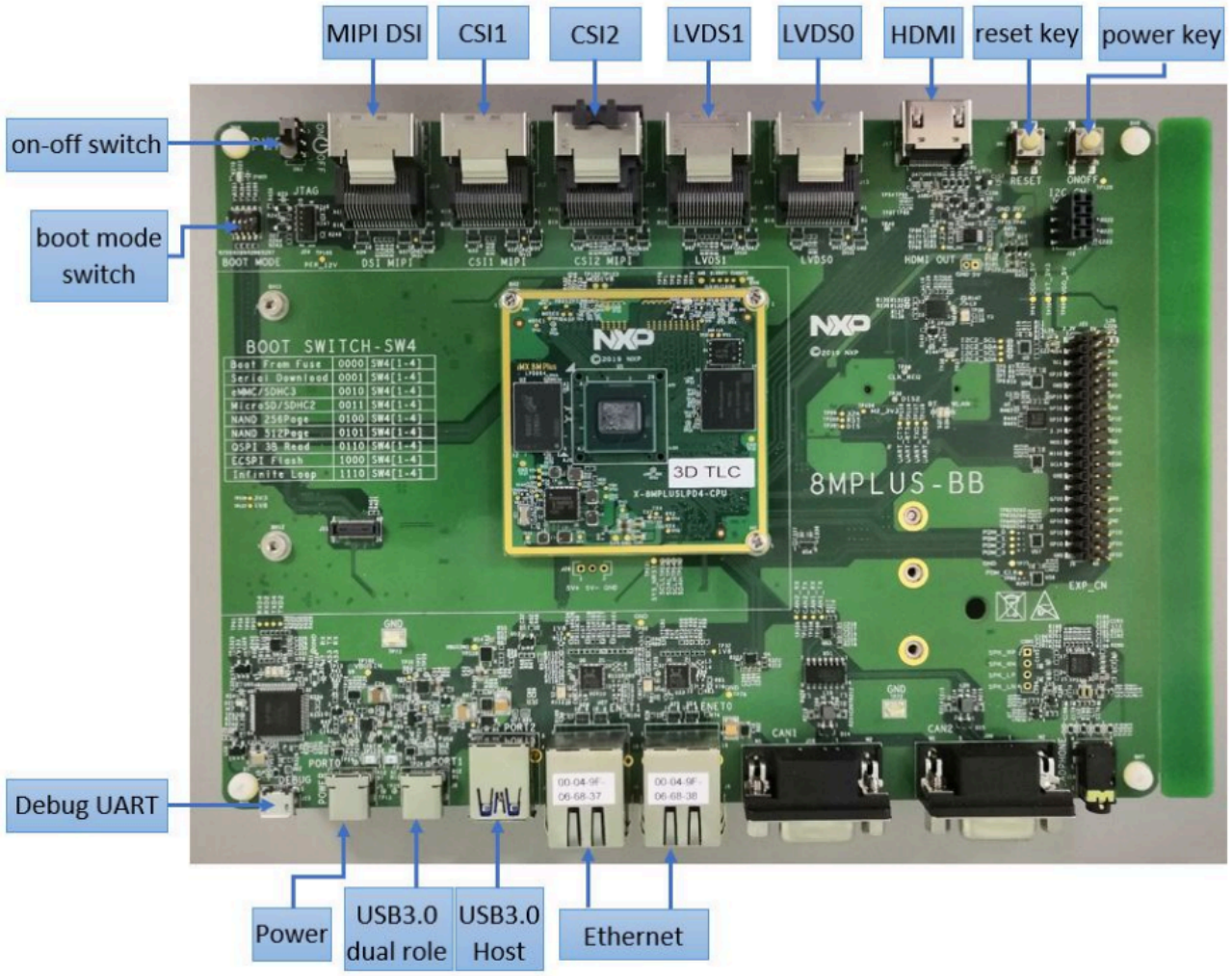


Figure 9. i.MX 8M Plus EVK board

Figure 10. i.MX mini SAS cable with LVDS-to-HDMI adapter

Figure 11. OV5640 MIPI CSI-2 camera

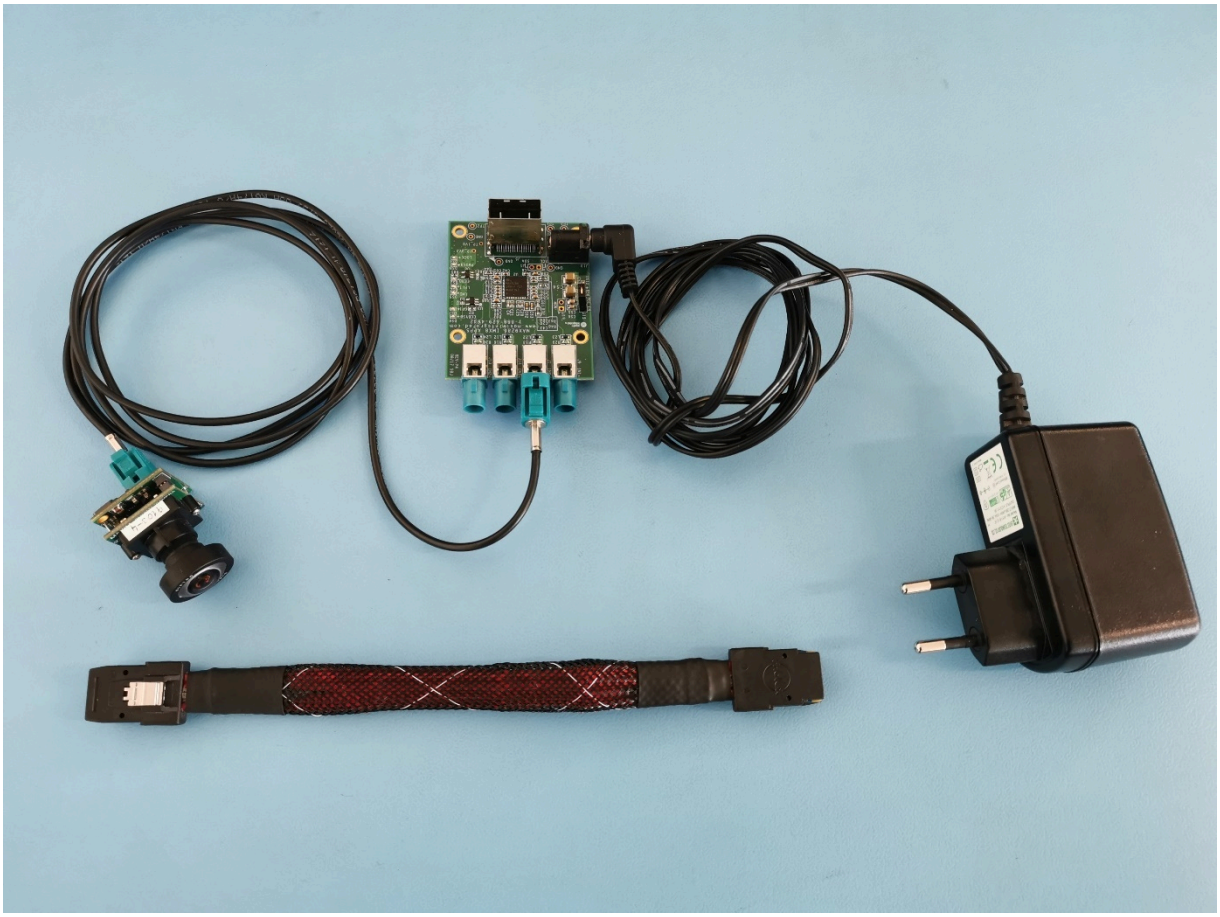


Figure 12. OV10635 Camera set

6.2 Display interface support

- An HDMI compatible display is supported and can be connected using a standard HDMI cable to the HDMI display interface on the EVK board (see Figure 9). This display interface is selected as the default one.
- The LVDS to HDMI converter IMX-LVDS-HDMI is supported. The converter is connected to the LVDS0 display interface on the EVK board (see Figure 9) with a mini-SAS cable (see Figure 10). Use a standard HDMI cable to connect an HDMI compatible display. It is necessary to update the Display0Interface parameter in the registry or `galcore.inf` to 0x4 and reboot/re-install the GPU driver (for more information, see i.MX Windows IoT User's Guide, IMXWGU).
- Optionally, a single LVDS display (connected to LVDS0) and a dual LVDS display (connected to both LVDS0 and LVDS1) can be used. It is necessary to update the Display0Interface parameter in the registry or `galcore.inf` to 0x4 (single LVDS) or 0x6 (dual LVDS) for the windows GPU driver and reboot/re-install the GPU driver (for more information, see i.MX Windows IoT User's Guide, IMXWGU).

- MIPI-DSI to HDMI converter IMX-MIPI-HDMI is supported. The converter is connected to the MIPI-DSI display interface on the EVK board (see Figure 9) with a mini-SAS cable (see Figure 2). Use a standard HDMI cable to connect an HDMI compatible display. It is necessary to update the Display2Interface parameter in the registry or `galcore.inf` to 0x2 and reboot/re-install the GPU driver (for more information, see i.MX Windows IoT User's Guide, IMXWGU).
- Optionally, a MIPI-DSI display (connected to MIPI-DSI) can be used. It is necessary to update the Display2Interface parameter in the registry or `galcore.inf` to 0x2 and reboot/re-install the GPU driver (for more information, see i.MX Windows IoT User's Guide, IMXWGU).

7 Working with the i.MX 8M Quad EVK Board

7.1 Board hardware

The figures below show different components of the i.MX 8M Quad EVK board.

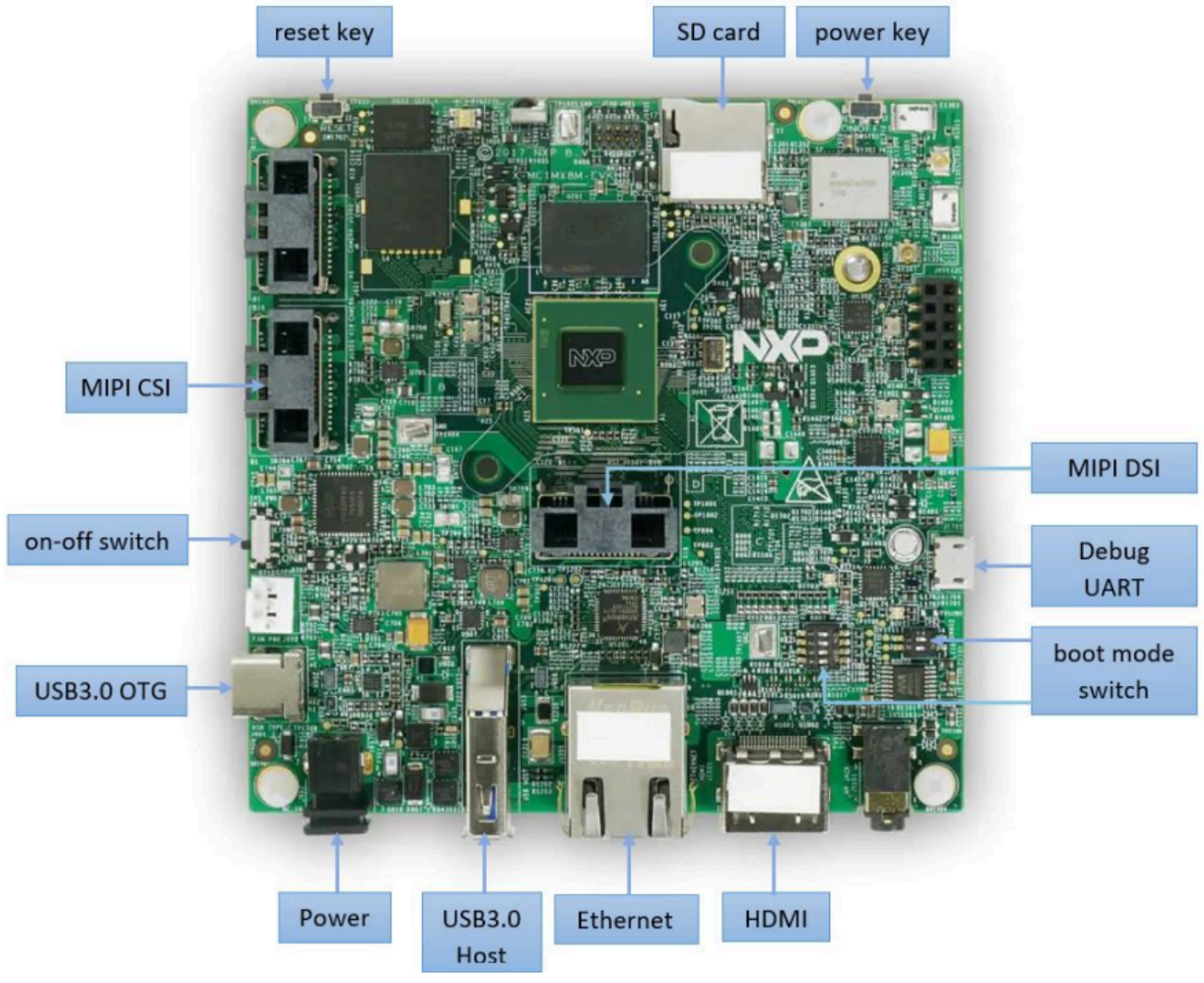


Figure 13. i.MX 8M Quad EVK board

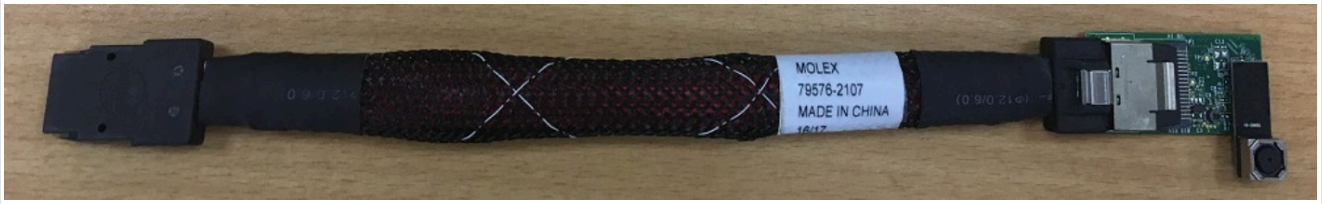


Figure 14. OV5640 MIPI CSI-2 camera

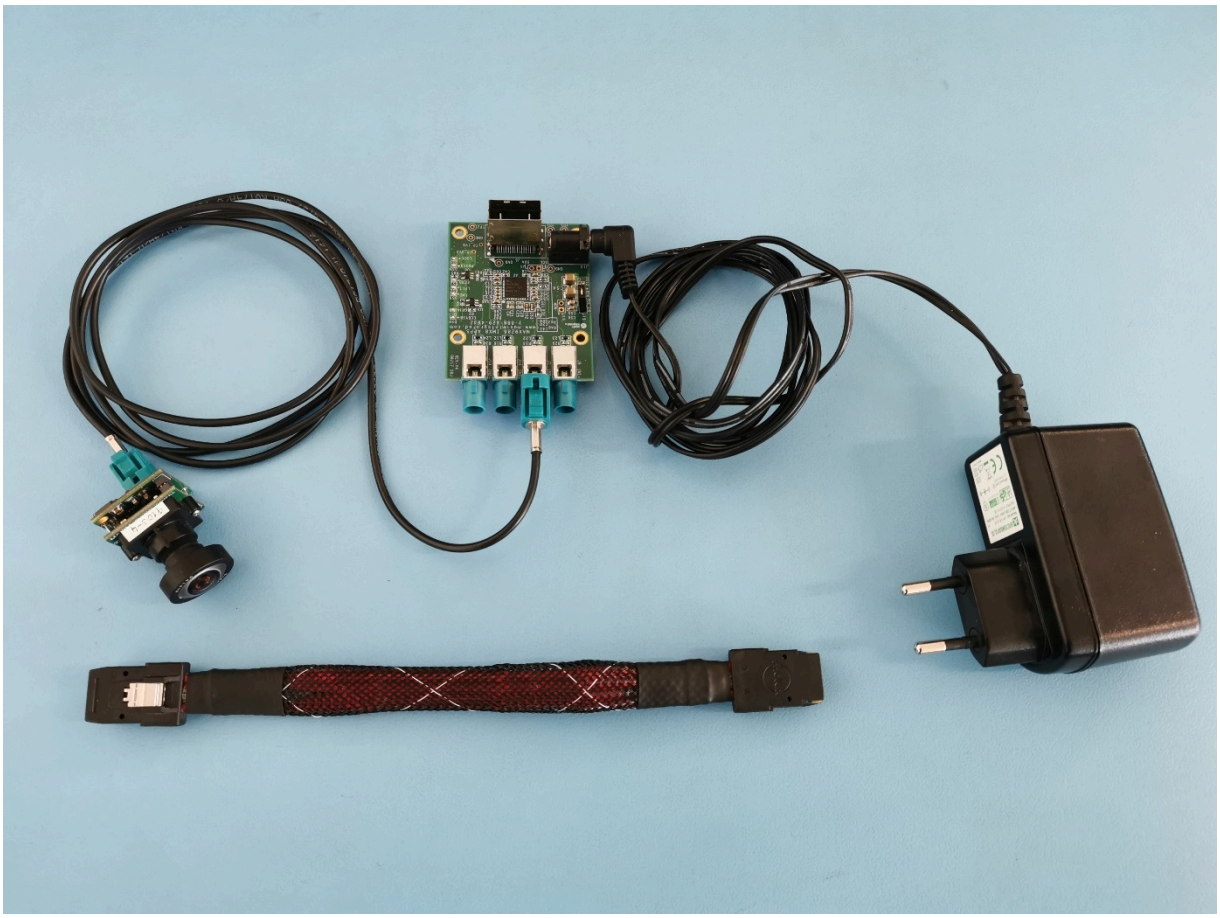


Figure 15. OV10635 Camera set

7.2 Display interface support

The HDMI interface is supported on the EVK board (see Figure 13). Use a standard HDMI cable to connect an HDMI compatible display.

8 Working with the i.MX 8QuadXPlus MEK Board

8.1 Board hardware

The figures below show different components of the i.MX 8QuadXPlus MEK board.

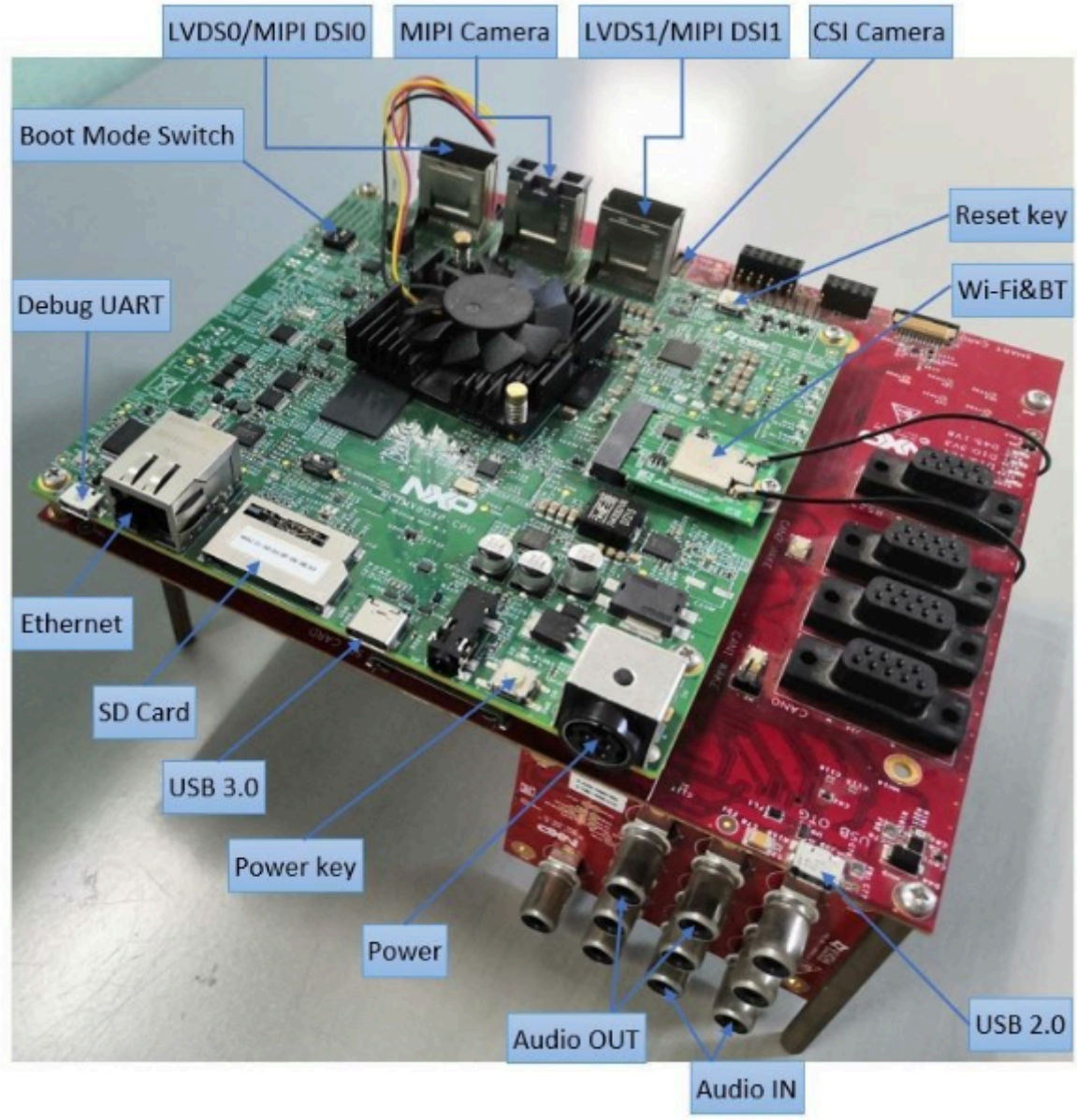


Figure 16. i.MX 8QuadXPlus MEK board

Figure 17. OV5640 MIPI CSI-2 camera

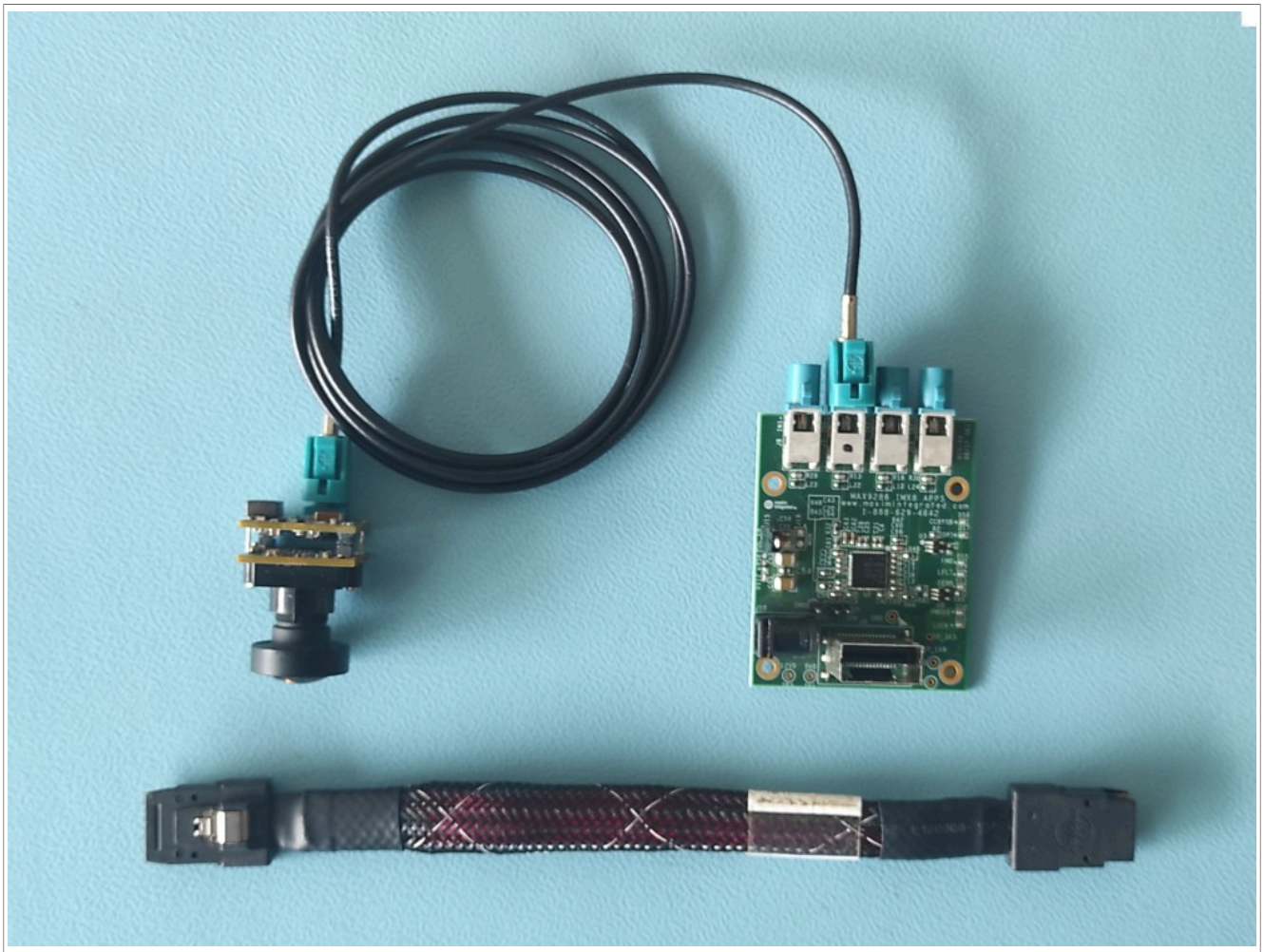


Figure 18. OV10635 Camera set

Note:

- Do not connect the 12 V external adapter to the camera plugged into the i.MX 8QuadXPlus MEK board. The 12 V power source is embedded on the development board.

8.2 Display interface support

- The LVDS to HDMI converter IMX-LVDS-HDMI is supported. The converter is connected to the LVDS0 display interface on the MEK board (see Figure 16) with a mini-SAS cable (see Figure 10). Use a standard HDMI cable to connect an HDMI compatible display.
- Optionally, an additional LVDS1 display interface can be used (LVDS to HDMI converter or LVDS display). It is necessary to update the Display1Interface parameter in the registry or `galcore.inf` to 0x5 and reboot/re-install the GPU driver (for more information, see i.MX Windows IoT User's Guide, IMXWGU).
- MIPI-DSI to HDMI converter IMX-MIPI-HDMI is supported. The converter is connected to the MIPI-DSI0/MIPI-DSI1 display interface on the EVK board (see Figure 16) with a mini-SAS cable (see Figure 2). Use a standard HDMI cable to connect an HDMI compatible display. It is necessary to update the Display0Interface (DSI0) or Display1Interface (DSI1) parameter in the registry or `galcore.inf` to 0x2 (DSI0) or 0x3 (DSI1) and reboot/re-install the GPU driver (for more information, see i.MX Windows IoT User's Guide, IMXWGU).
- Optionally, a MIPI-DSI display (connected to MIPI-DSI0/MIPI-DSI1) can be used. It is necessary to update the Display0Interface (DSI0) or Display1Interface (DSI1) parameter in the registry or `galcore.inf` to 0x2 (DSI0)

or 0x3 (DSI1) and reboot/re-install the GPU driver (for more information, see i.MX Windows IoT User's Guide, IMXWGU).

9 Working with the i.MX 93 EVK Board

9.1 Board hardware

The figures below show different components of the i.MX 93 EVK board.

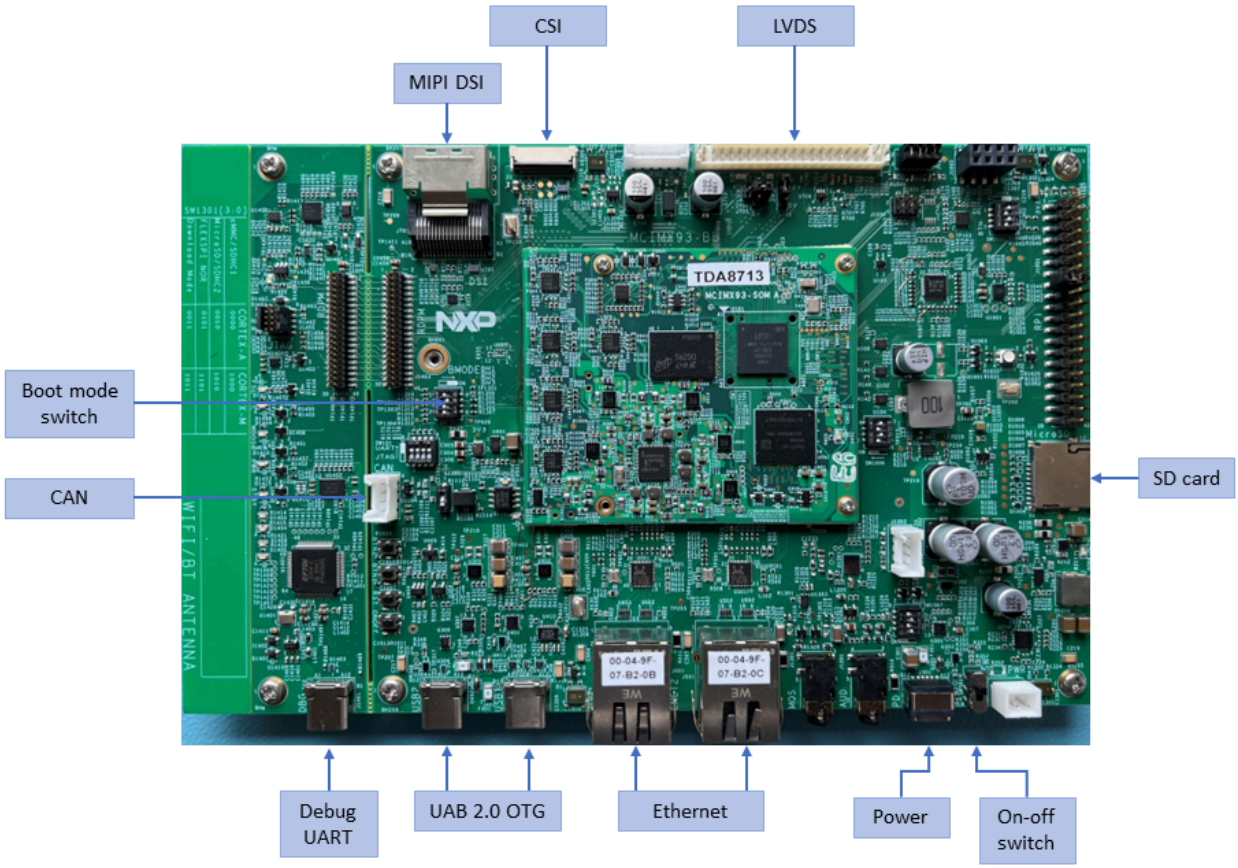


Figure 19. i.MX 93 EVK board

Figure 20. i.MX Mini SAS cable with the DSI-to-HDMI adapter IMX-MIPI-HDMI

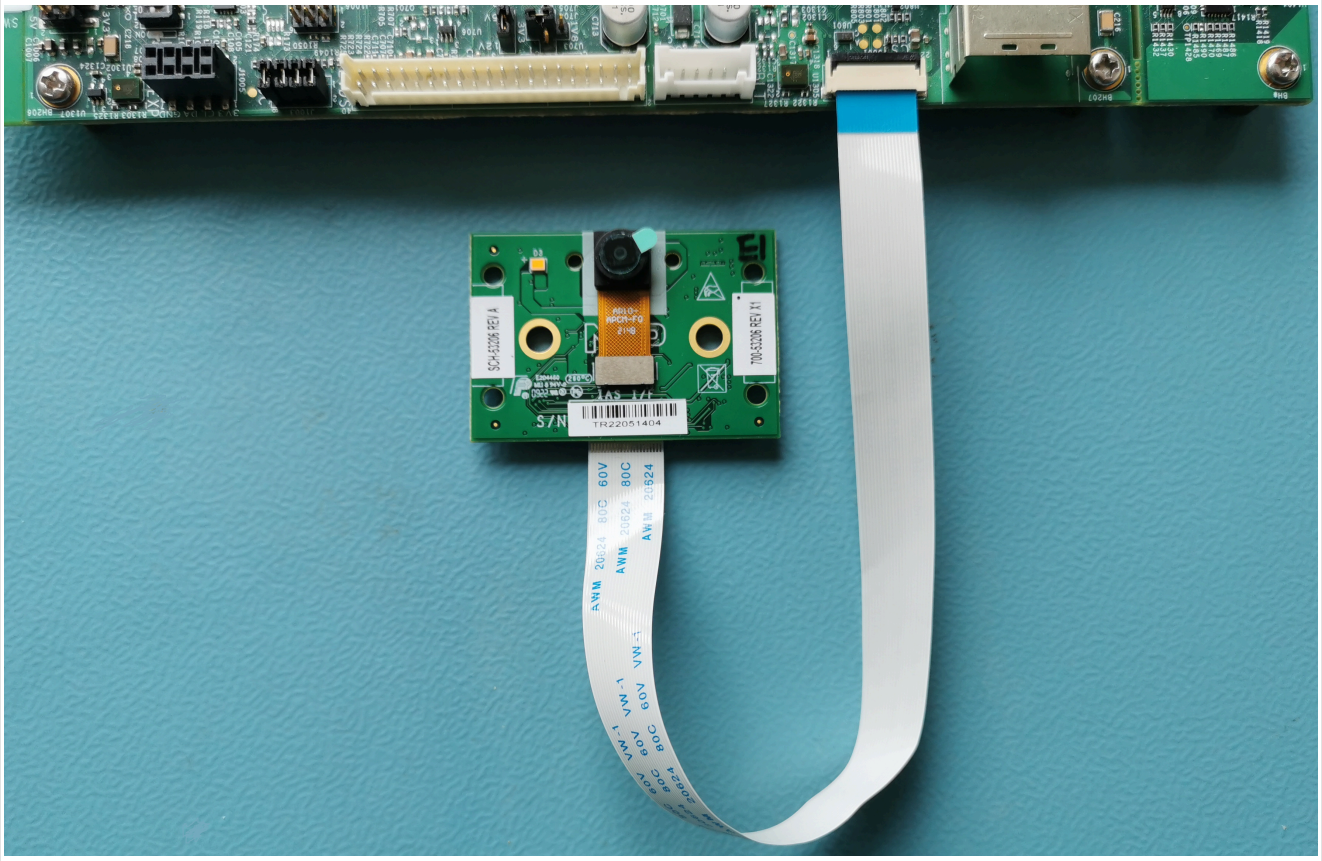


Figure 21. RPI-CAM-MIPI camera board

9.2 Display interface support

The MIPI-DSI to HDMI converter IMX-MIPI-HDMI is supported as the default display interface. The converter is connected to the MIPI-DSI display interface on the EVK board (see Figure 19) with a mini-SAS cable (see Figure 20). Use a standard HDMI cable to connect an HDMI compatible display. Alternatively, an LVDS panel EV121WXM-N12 with resolution 1280x800 can be connected to the LVDS display interface on the EVK board (see Figure 22). To switch to the LVDS display interface, set `PcdDisplayInterface` to 2 in the `/mu_platform_nxp/NXP/MX93_11X11_EVK/MX93_11X11_EVK.dsc` file. The firmware must then be recompiled.

Figure 22. EV121WXM-N12 LVDS panel

10 Flashing Windows IoT image

Windows IoT must be installed on the eMMC from the SD card. Preparing the bootable SD card consists of two steps: flashing the Windows IoT installer and flashing the firmware to an SD card.

10.1 Flashing Windows IoT Installer to the SD card

Currently, the only way to deploy a Windows IoT Enterprise on the onboard eMMC is to use WinPE.

OS (Windows Preinstallation Environment) to write the Windows IoT image to eMMC.

Windows manufacturing OS (WinPE) that can be fully loaded and run from memory without using persistent storage. The following steps create an SD card with WinPE and a Windows IoT image that contains the BSP drivers. The boot firmware checks the SD card and boots WinPE, which then installs the Windows IoT image to the eMMC.

1. Decompress the W21H2-1-x-x-imx-windows-bsp-binaries.zip file. The package contains release-signed prebuilt binaries and image files.
2. Open the elevated command prompt and navigate to the *IoTEntOnNXP*directory.
3. Mount the previously downloaded Windows IoT Enterprise ISO image file (see chapter Software requirements) and copy the *install.wim* file from the <*DVD mount drive:*>*sources**install.wim*to the *IoTEntOnNXP* directory.
   Note: This step can be skipped - instead of mounting the ISO image file, copy it into the *install.wim* directory and add parameter `/iso` into the command line in the next step.
4. If you are building an image with Windows IoT Enterprise, execute the command below and then skip to step 6. If you are building an image with Windows 11 IoT Enterprise, skip this step and go to step 5:
   `.\make-winpe-enterprise.cmd /disable_updates /test_signing`
   This command creates a copy of the selected *install.wim* image with injected i.MX drivers and applied updates from the *kbpatch/* directory. These patches are for Windows 21H2, build 19044.1288 and update the image to build 19044.2566.
   Note: Be sure to copy the whole command line.
5. If you are building an image with Windows 11 IoT Enterprise, execute the command:
   `.\make-winpe-enterprise.cmd /disable_updates /test_signing /no_patch`
   This command creates a copy of the selected install.wim image with injected i.MX drivers but does NOT apply any updates from the kbpatch/ directory
6. Execute the command:
   `.\make-winpe-enterprise.cmd /apply <disk_number>`
   where <*disk_number*> is the physical number of the SD card disk on your PC. It can be obtained using the Disk Management tool (right-click the start menu icon and select Disk Management).
   This command deploys the WinPE image to the SD card.
   CAUTION:
   Make sure to select the correct disk number, as this step formats the selected disk! The WinPE-based Windows installer is now deployed on the SD card.
7. Continue with the firmware installation to the SD card.

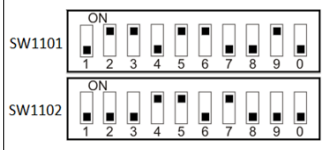
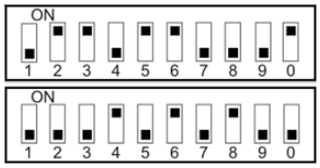
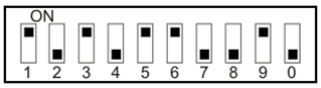
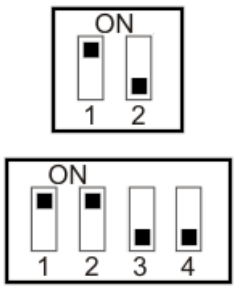
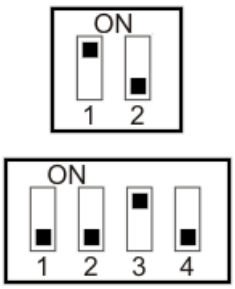
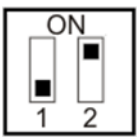
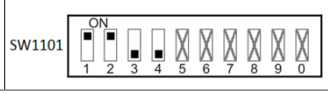
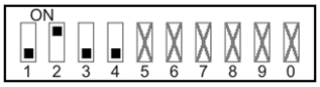
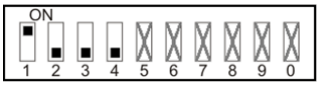
10.2 Flashing firmware to the SD card

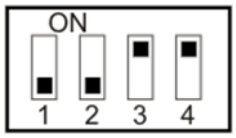
During active development of the boot firmware, it can be time-consuming and error-prone to repeatedly change the dip switches between UUU download mode and eMMC boot mode. To simplify this process, i.MX EVK boards support SD card boot mode that allows you to keep the boot firmware on an SD card.

To deploy boot firmware to an SD card from Windows, use the Cfimager tool.

Perform the following steps to flash the firmware to the SD card:

1. Download the NXP cfimager tool and copy it into the *firmware* directory or a directory listed in the system environment variable %PATH%.
2. Navigate to the *firmware* directory.
3. Plug the SD card into the host PC and execute the following board-specific command:
   For the i.MX 8M Mini EVK board:
   ```
   .\flash_bootloader.cmd /device MX8M_MINI_EVK /target_drive <SD card driver letter, for example, f:>
   ```
   For the i.MX 8M Quad EVK board:
   ```
   .\flash_bootloader.cmd /device MX8M_EVK /target_drive <SD card driver letter, for example, f:>
   ``
   ```
   For the i.MX 8M Nano EVK board:
   ```
   .\flash_bootloader.cmd /device MX8M_NANO_EVK /target_drive <SD card driver letter, for example, f:>
   ``
   ```
   For the i.MX 8M Plus EVK board:
   ```
   .\flash_bootloader.cmd /device MX8M_PLUS_EVK /target_drive <SD card driver letter, for example, f:>
   ``
   ```
   For the i.MX 8QuadXPlus MEK board:
   ```
   .\flash_bootloader.cmd /device MX8QXP_MEK /target_drive <SD card driver letter, for example, f:>
   ```
   For the i.MX 93 EVK board:
   ```
   .\flash_bootloader.cmd /device MX93_11X11_EVK /target_drive <SD card driver letter, for example, f:>
   ```
4. Power off the board.
5. Insert the SD card to the board.
6. Change the boot device to the SD card.
7. Power on the board.

Board	SD card boot	eMMC boot	USB download mode
MX8M_MINI_EVK			
MX8M_EVK			
MX8M_NANO_EVK			

Board	SD card boot	eMMC boot	USB download mode
MX8M_PLUS_EVK			
MX8QXP_CPU			
MX93_EVK			

10.3 Flashing Firmware to eMMC

The second option is to load the firmware to the eMMC memory. To flash the bootable firmware to eMMC, we recommend downloading the UUU \(Universal Update Utility\) 1.5.21 tool used for BSP testing from UUU (Universal Update Utility) tool.

Note: The UUU tool requires U-Boot with UUU boot support enabled. For security reasons, it is recommended to disable UUU download support in the U-Boot in final designs.

To deploy boot firmware to eMMC via USB download mode, follow the steps below:

1. Navigate to the firmware directory. In the source package, it is in `imx-windows-iot/BSP/firmware`.
2. Download the UUU tool and copy it into the `firmware\tools\uuu\` directory (Create the `tools\` and `uuu\` directories, as they are not present in the source package).
3. Start preparing the board by turning it off.
4. Switch the target board to USB download mode using DIP switches. See the table in section Flashing firmware to the SD card.
5. Connect a USB-C cable to the desktop and the board's USB-C port. Then turn on the board.
6. Run `flash_bootloader.cmd` as: `.\flash_bootloader.cmd /device <device_name> /secure <insecure=0, secure=1>` where tag `/secure` specifies temporary firmware with the UUU tool support (by default `firmware_uuu.bin` is used for nonsecure selection, `signed_firmware_uuu.bin` is used for secure build selection). This firmware is used to program the user's firmware on eMMC. Ensure that the corresponding firmware is built. If the corresponding firmware is not found, the user is prompted to select a custom firmware or exit the program.
7. When the process is complete, power off the board. Disconnect the USB-C cable and connect the board to the PC. Failure to disconnect from the PC may cause errors in subsequent steps.
8. Switch the board to eMMC boot mode. See the table in section Flashing firmware to the SD card. Basic Board Setup UUU documentation is available at nxp.com.

10.4 Installing Windows IoT on eMMC and Booting Windows IoT

1. Power on the board and let the Windows installer install Windows IoT Enterprise to the eMMC, then reboot into installed Windows. Installing Windows IoT Enterprise from the SD card can take around 30 minutes. **Note:** The WinPE installer renames the EFI folder at the root of the SD card to _efi, which causes UEFI to skip the SD card at boot time. This allows you to keep the SD card inserted across reboots without having

Windows IoT reinstalled on each reboot. If you wish to boot into the WinPE installer again, you can rename _efi back to EFI.

2. Make sure to disable sleep in “Power and sleep” settings after Windows OS boots up to avoid unexpected system hangs.

11 Working with cameras

Any MIPI CSI-2 camera module has to be configured in ACPI first. This requires modification and rebuild of boot firmware, particularly UEFI, as described here. Before starting, examine the table below for supported combinations of the NXP development boards and external camera modules.

11.1 Camera Connection and UEFI Parameters

The table below lists the camera type with the label of the MIPI CSI-2 connector and the PCD Symbol is used to enable this configuration within ACPI.

MX8M_EVK Board:

Camera	Connector	UEFI PCD symbol/macro
OV5640	J1502 (CAMERA)	PcdCsi1CameraOv5640
OV5640	J1503 (CAMERA)	PcdCsi2CameraOv5640
OV10635	J1502 (CAMERA)	PcdCsi1CameraOv10635
OV10635	J1503 (CAMERA)	PcdCsi2CameraOv10635

MX8M_PLUS_EVK Board:

Camera	Connector	UEFI PCD symbol/macro
OV5640	J12 (CSI 1 MIPI)	PcdCsi1CameraOv5640
OV5640	J13 (CSI 2 MIPI)	PcdCsi2CameraOv5640
OV10635	J12 (CSI 1 MIPI)	PcdCsi1CameraOv10635
OV10635	J13 (CSI 2 MIPI)	PcdCsi2CameraOv10635

MX8M_NANO_EVK Board:

Camera	Connector	UEFI PCD symbol/macro
OV5640	J802 (CSI MIPI)	PcdCsi1CameraOv5640
OV10635	J802 (CSI MIPI)	PcdCsi1CameraOv10635

MX8M_MINI_EVK Board:

Camera	Connector	UEFI PCD symbol/macro
OV5640	J802 (CSI MIPI)	PcdCsi1CameraOv5640
OV10635	J802 (CSI MIPI)	PcdCsi1CameraOv10635

MX8QXP_CPU Board:

Camera	Connector	UEFI PCD symbol/macro
OV5640	J2 (CSI)	PcdCsi1CameraOv5640
OV10635	J2 (CSI)	PcdCsi1CameraOv10635

MX93_11x11_EVK Board:

Camera	Connector	UEFI PCD symbol/macro
RPI-CAM-MIPI (AP1302 ISP + AR0144)	J801 (CSI)	PcdCsi1CameraRpiCamMipi

The MX93_11x11_EVK cable orientation warning.

Be careful with the correct orientation of the ribbon cable to interconnect the RPI-CAM-MIPI with MX93_11x11_EVK. The blue bars on the ribbon cable must be oriented as depicted bellow in Figure 23 and Figure 24.

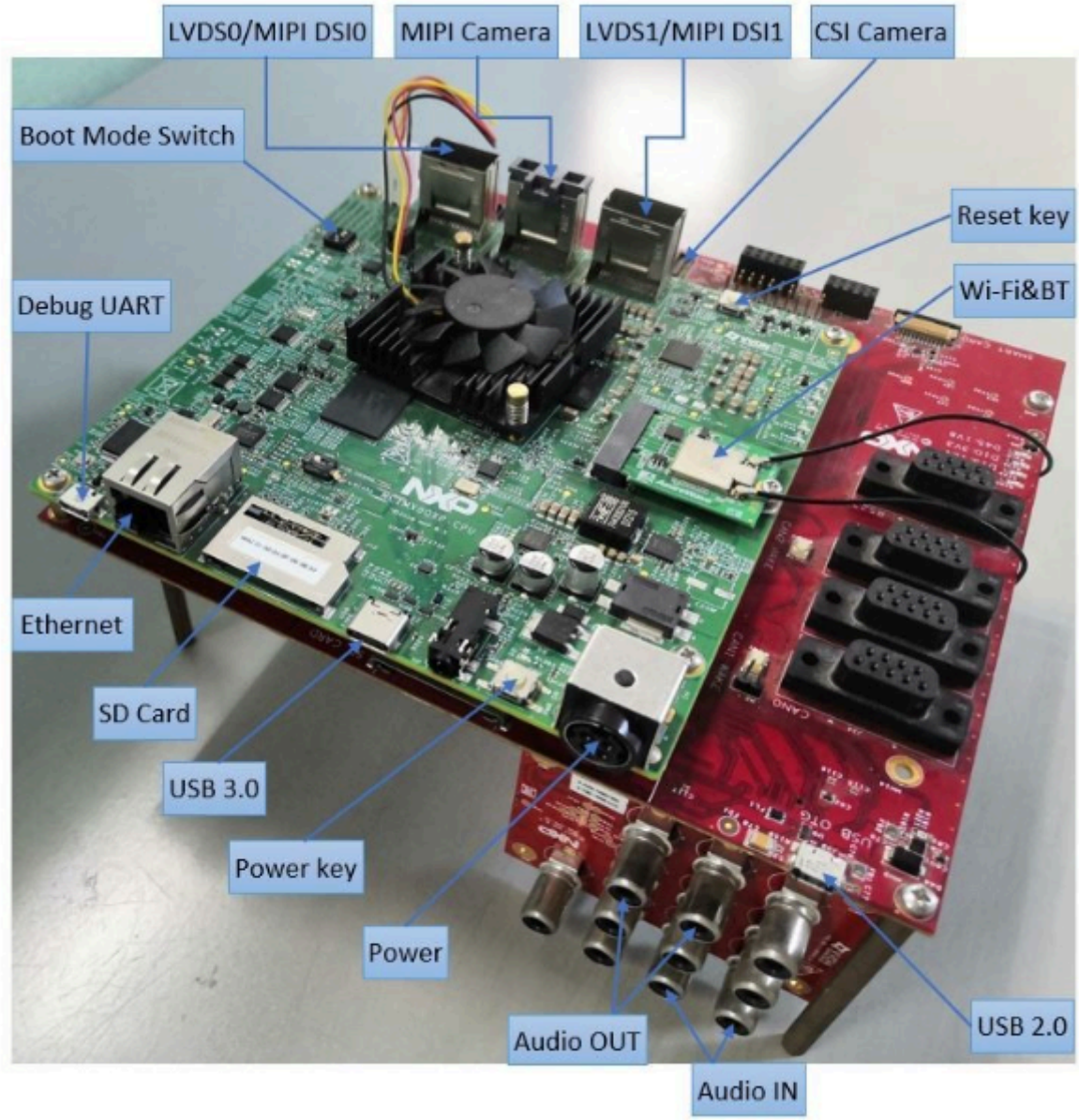


Figure 23. i.MX 8QuadXPlus MEK board

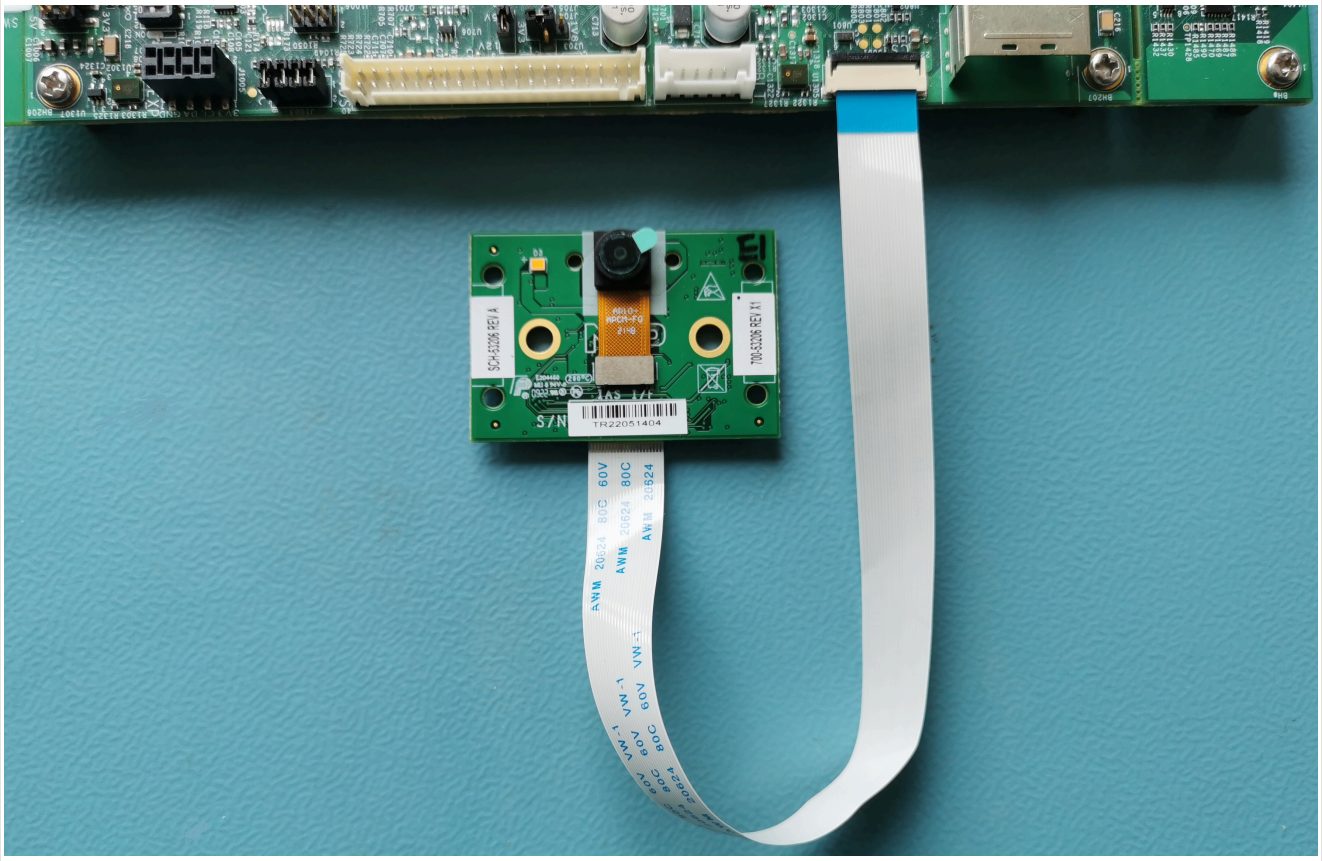


Figure 24. Connection of the ribbon cable on MX93_11x11_EVK board side

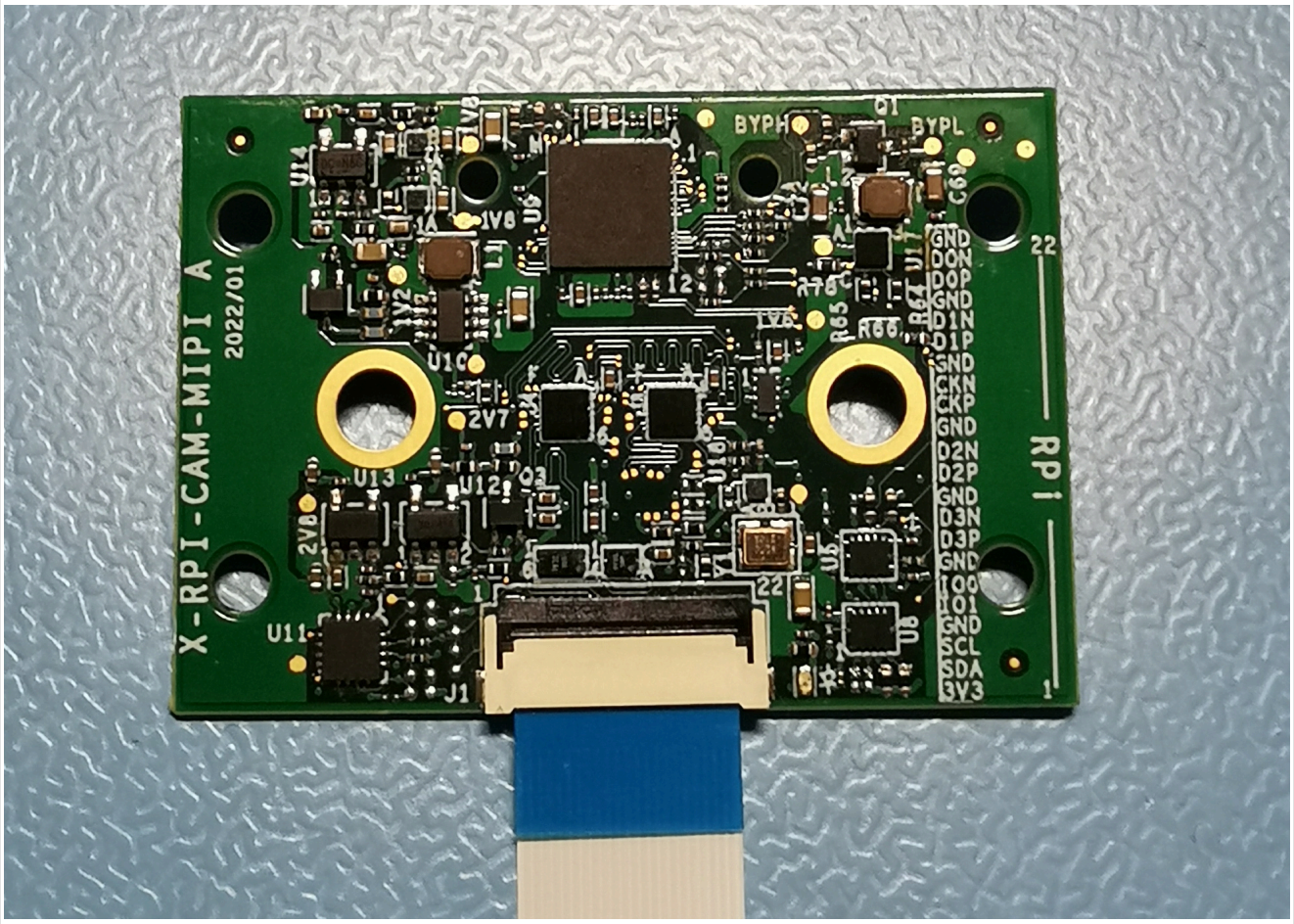


Figure 25. Connection of the ribbon cable on X-RPI-CAM-MIPI camera board side

11.2 Enabling Camera in ACPI

1. Prepare boot firmware for compilation as described in Windows IoT User's Guide (IMXWUG).
2. Having downloaded BSP sources, you can navigate to the `mu_platform_nxp/NXP/<board_name>/<board_name>.dsc` file and open it in a text editor. This file defines values of macros for enabling and disabling the cameras.
   Example: The Dsc file for i.MX 8M EVK board is located here:

   ```
   win10-iot-bsp\mu_platform_nxp\NXP\MX8M_EVK\MX8M_EVK.dsc
   ```

3. In this file, find and configure the `giMX8TokenSpaceGuid.PcdCsi<N>Camera<name>` macros.
   Example: i.MX 8M EVK with OV10635 camera in CSI1 interface requires:

   ```
   1. Setting `giMX8TokenSpaceGuid.PcdCsi1CameraOv5640` to zero
   2. Setting `giMX8TokenSpaceGuid.PcdCsi1CameraOv10635` to one.
   ```

4. Having saved your modifications to the dsc file, you can compile the firmware as described in the Windows IoT User's Guide (IMXWUG).
5. Flash the compiled firmware to an SD card or eMMC.
6. Power off the board and connect the camera. Connecting the camera without powering off the board will damage it.
7. After booting Windows on the development board, the camera should be available in the device manager and ready for use.

12 Revision history

Revision number	Date	Substantive changes
IMXWQSG_Rev.1.5.2	11 November 2024	Minor technical changes.
IMXWQSG_Rev.1.5.1	24 May 2024	Minor technical changes.
IMXWQSG_Rev.1.5.0	12 August 2023	Minor technical changes.
IMXWQSG_Rev.1.4.1	31 July 2023	Minor technical changes.
IMXWQSG_Rev.1.4.0	20 March 2023	i.MX 93 EVK board support is added.
IMXWQSG_Rev.1.3.0	20 December 2022	i.MX 8QuadXPlus MEK board support is added.
IMXWQSG_Rev.1.2.1	26 October 2022	Section 3 is updated.
IMXWQSG_Rev.1.2.0	23 September 2022	Initial version of the Quick Start Guide.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents