

UG10110

Getting Started with NavQPlus Companion Computer

Rev. 1.0 — 14 May 2024

User guide

Document information

Information	Content
Keywords	UG10110, NavQPlus, Companion Computer, NavQPlus Companion Computer, NavQ+, 8MPNAVQ
Abstract	8MPNAVQ or NavQPlus is a small purpose-built Linux computer evaluation kit (EVK) based on the NXP i.MX 8M Plus System-on-Chip (SoC). It focuses on the common needs of mobile robotics systems, with a small form factor, Dronecode-compliant JST-GH connectors, and an available software stack including Ubuntu Linux and ROS2.



1 Introduction

8MPNAVQ or NavQPlus is a small purpose-built Linux computer evaluation kit (EVK) based on the [NXP i.MX 8M Plus System-on-Chip \(SoC\)](#). It focuses on the common needs of mobile robotics systems, with a small form factor, [Dronecode](#)-compliant JST-GH connectors, and an available software stack including Ubuntu Linux and [ROS2](#).

The complete design is available for companies building their own similar hardware. NavQPlus is built as a stack of boards. The top board being a SoM (System-on-Module) containing the processor, memory, and other components with strict layout requirements. The secondary boards are relatively inexpensive, often four-layer boards, and allow for versions with customization to be easily built.

Note:

The SoM is almost identical to the larger [NXP EVK for i.MX8M Plus](#) except for the I/O voltage level being changed to 3.3 V. This makes NavQPlus an excellent stepping stone or bridge from large EVK to a system that can be duplicated for testing in situ, or even copied directly for your application.

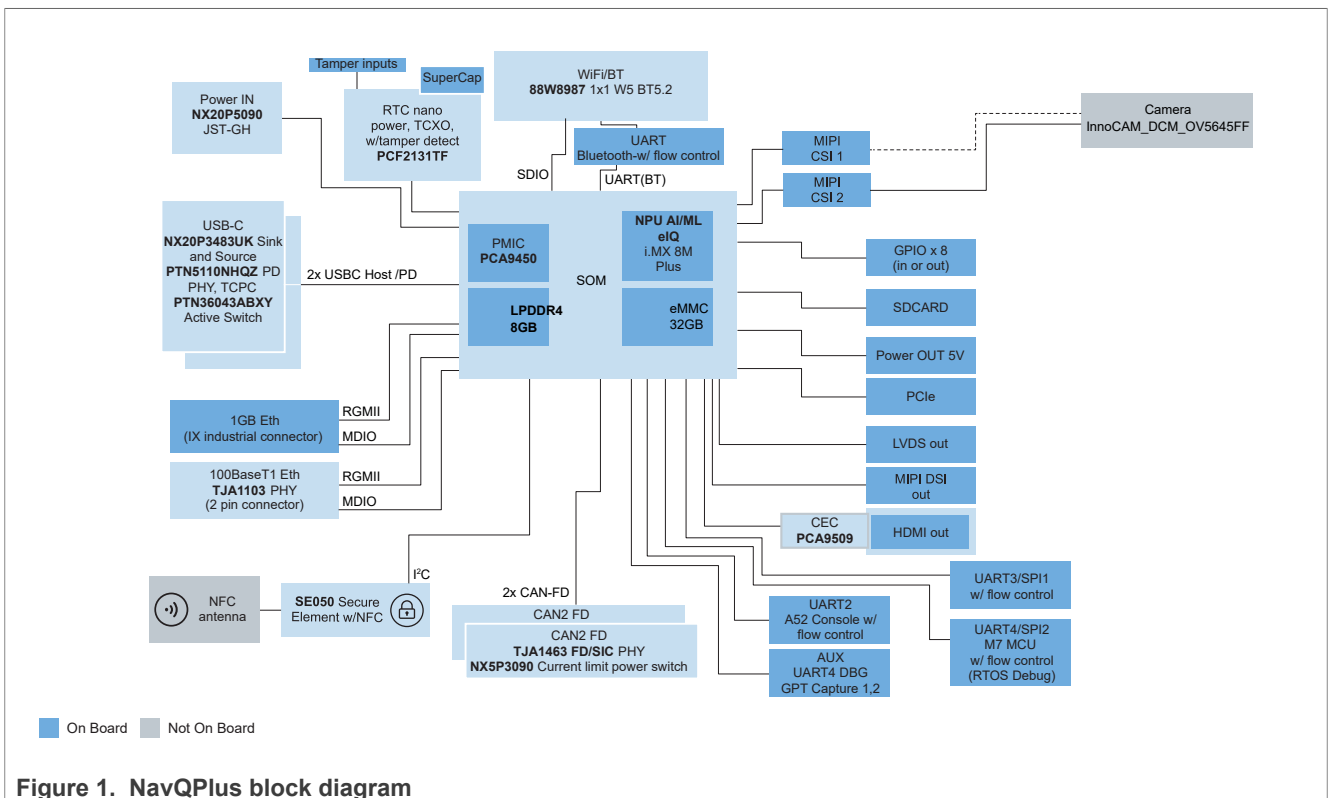


Figure 1. NavQPlus block diagram

Refer to the NXP.com landing page <https://www.nxp.com/navqplus> for NavQPlus for availability, pricing, availability, and distribution channels.

Note: Currently, there are two variants of the NavQPlus available. The part numbers start with 8MPNAVQ. The variants are:

- **8MPNAVQ-8GB-XG** This variant does not include the IX Industrial Gigabit Ethernet Connector and PHY on board. All other components are installed, including 100BaseT1 and WiFi. Ethernet over USB-C gadget mode is also available.
- **8MPNAVQ-8GB-G** This variant includes the IX industrial connector and Gigabit Ethernet PHY.

1.1 Features

NXP i.MX 8M Plus SoC on a SoM with low-power double data 4 rate dynamic random-access memory (LPDDR4 DRAM) and eMMC flash.

- 4x Arm Cortex-A53 core
- 1x Arm Cortex-M7 core
- 1x Neural Processing Unit (2.3 TOPS)
- 1080p60 H.265/H.264 encoder
- Dual Camera Image Signal Processor (HDR, Dewarp)

A secondary board with connectors to hardware interfaces, such as:

- Dual MIPI-CSI camera interfaces
- Two CAN-FD interfaces
- I2C, SPI, UART, GPIO
- SD card slot
- 2.4/5GHz WiFi and Bluetooth 5.0 using [NXP 88W8987](#)-based [Murata Type 1ZM module](#)
- Micro-HDMI, MIPI-DSI, and LVDS for displays
- USB-C, including power input and output
- 1 Gigabit Ethernet with ix Industrial connector
- JTAG BOOT

1.2 Applications

NavQPlus is suitable for many purposes, including generic robots, various vision systems, and AI/ML applications.

- Unmanned aerial vehicles (UAVs)
 - Such as multicopters and vertical take-off and landing (VTOL) air crafts.
- Rovers and other unmanned ground vehicles (UGVs)
 - Road-going delivery vehicles
 - Robotic lawn mowers
 - Robotic vacuum cleaners
- Marine vessels
- Camera and vision processing modules
- Time-of-flight (ToF) cameras
- AI/ML inference
- Cellular gateway
- Vision systems in other applications
 - For example, a hospital bed monitor that detects if a patient is sitting up or at risk of falling out of bed.

Several complete developer tool examples are the [NXP HoverGames drone](#), MR-B3RB Mobile Robotics Bugg, and the [NXP Cup car](#).

1.3 Software

The intent of the 8MPNavQ enable developers with a solution that allows them to harness common robotics packages and libraries such as:

- ROS/ROS2
- OpenCV

- GStreamer
- pyelQ
 - TensorFlow/TFLite
 - PyTorch
 - Arm NN

The 8MPNavQ runs Linux with a package manager enabling the installation of almost all the packages that require to complete your projects successfully and efficiently.

2 Caution and disclaimers

Refer to the HoverGames disclaimer below when using NavQPlus with a drone or similar vehicle.

https://app.gitbook.com/o/-L9GLsni4p7csCR7QCJ8/s/-L9GLtb-Tz_XaKbQu-AI/disclaimer.

Note: *It is an assumption that as a NavQPlus user, you know how to operate an embedded Linux computer in a headless terminal environment. While NavQPlus can support a desktop environment, it is **not** a personal computer or a desktop computer. Therefore, many graphical applications designed for these devices do **not** work due to the requirement of device-specific libraries and graphics drivers.*

2.1 Resources

If you are not comfortable operating a Linux computer in a terminal environment, refer to the following resources for help.

- [Linux command line for beginners](#)
- [Embedded Linux introduction](#)

2.2 Federal Communication Commission and Industry Canada information

This topic provides information on the Federal Communication Commission (FCC) and Industry Canada (IC) information.

2.2.1 FCC information

Contains the Transmitter Module FCC ID: VPYLB1ZM or Contains FCC ID: VPYLB1ZM

CAUTION: *Changes or modifications not explicitly approved by the party responsible for compliance could void the user authority to operate the equipment.*

Important: *Do not colocate or operate the transmitter with any other antenna or transmitter.*

This device complies with part 15 of the FCC rules. Operation is subject to the following conditions.

- This device may not cause harmful interference.
- This device must accept any interference received, including interference that causes undesired operation.

The available scientific evidence does not show that any health problems are associated with using low-power wireless devices. However, there is no proof that these low-power wireless devices are safe.

Low-power wireless devices emit low levels of radio frequency energy (RF) in the microwave range when in use. However, high levels of RF can produce health effects by heating tissue. No known adverse effect of exposure to low-level radio frequency has been discovered. Many studies of low-level RF exposures have not found any biological effects. Some studies have suggested that some biological effects can occur. However, there are no confirmed findings. Tests indicate that LBEE5QD1ZM complies with FCC radiation exposure limits set for an uncontrolled environment and meets the FCC Radio Frequency (RF) Exposure Guidelines.

It is necessary to take a specific absorption rate (SAR) test with your set mounting this module, except for Bluetooth. Class 2 permissive change application is necessary using the SAR report. For details, contact Murata.

This equipment complies with FCC radiation exposure limits set for an uncontrolled environment and meets the FCC Radio Frequency (RF) Exposure Guidelines. Ensure to install and operate the equipment keeping the radiator at least 20 cm or more away from your body.

2.2.2 Canadian IC information

Contains IC: 772C-LB1ZM. Operation is subject to the following conditions.

- This device may not cause interference.
- This device must accept any interference, including interference that may cause undesired operation of the device.

The software passed down through the MAC initiates the data transmission. The transmission happens through the digital, and analog baseband, and finally to the radio frequency (RF) chip. The media access controller (MAC) initiates several special packets. The above are the only ways that the digital baseband portion turns ON the RF transmitter, which then turns OFF at the end of the packet. Therefore, the transmitter is ON only when one of the aforementioned packets transmits. In other words, this device automatically discontinues the transmission in the absence of information to transmit or an operational failure.

This radio transmitter (IC Number: 772C-LB1ZM) identifies the device by certification number or model number. For Category 2, Industry Canada approves the operation with the antenna types listed below with the maximum permissible gain indicated. Prohibit the use of the device with antenna types not listed having a gain greater than the maximum gain indicated for that type.

- 146153 Dual Dipole antenna Gain: +3.2 dBi@2.4 GHz + 4.25 dBi@5 GHz
- 146187 Dual Dipole antenna Gain: +3.4 dBi@2.4 GHz + 4.75 dBi@5 GHz
- LBEE5QD1ZM-Antenna monopole antenna Gain: +3.6 dBi@2.4 GHz + 4.6 dBi@5 GHz

The available scientific evidence does not show that any health problems are associated with using low-power wireless devices. However, there is no proof that these low-power wireless devices are safe. Low-power wireless devices emit low levels of radio frequency energy (RF) in the microwave range when in use. However, high levels of RF can produce health effects by heating tissue. There are no known adverse health effects of exposure to top-level RF that does not produce heating effects. Many studies of low-level RF exposures have not found any biological effects. Some studies have suggested that some biological effects can occur. However, there are no confirmed findings. Tests indicate that LBEE5QD1ZM complies with the IC radiation exposure limits set for an uncontrolled environment. It also meets RSS-102 of the IC radio frequency (RF) Exposure rules.

It is necessary to take a specific absorption rate (SAR) test with your set mounting this module.

Class 4 permissive change application is necessary using the SAR report. For details, contact Murata.

This equipment complies with IC radiation exposure limits set for an uncontrolled environment and meets the RSS-102 of the IC radio frequency (RF) Exposure rules. Ensure to install and operate the equipment keeping the radiator at least 20 cm or more away from your body.

3 Prerequisites

The NavQPlus is an embedded Linux device. You need terminal access via a PC or Laptop to interface with the board. While a Windows PC can have terminal programs, working in a Linux host environment allows flexibility, capability, and ability to cross-compile software. Most examples used in the section assume that there is a Linux host PC running Ubuntu 22.04 or later attached.

3.1 Ubuntu desktop

For details on how to install the Ubuntu desktop on a development computer, refer to <https://ubuntu.com/download>.

3.2 Supported hardware

NavQPlus is a standalone embedded Linux device. There are many peripherals that attach and interface with it in the presence of the correct drivers. In addition, the Mobile Robotics teams with several complimentary devices and boards, which work with the NavQPlus.

- MR-B3RB (Buggy3 RevB): a robotic platform in the form of an Ackerman steering small car.
- iRobot Create3: AKA Turtlebot 4, ROS development platform.
- RDDRONE-T1ADAPT: Adapter board from 100BaseT1 to RJ45.
- MR-T1ETH8: T1+RJ45 Ethernet switch.
- MR-CANHUBK344: MCU board with S32K344, 6 CAN, T1 Ethernet.
- CANHUB-ADAP: plug on board with IMU and GPS connectors.
- MR-VMU-RT1176.
- Pixhawk V6X standard Vehicle Management Unit, with IMU.
Note: This platform supports Both NuttX and Zephyr.
- RDDRONE-BMS772: Battery Management 6S.
- UCANS32K146: CAN node board.
- HDMI displays; Many off the shelf small displays.

3.3 Supported software

This section provides information on [Section 3.3.1 "NavQPlus enablement"](#), [Section 3.3.2 "Open source community"](#), and [Section 3.3.3 "Yocto Linux"](#).

3.3.1 NavQPlus enablement

NavQPlus has specific enablement that is not available in generic EVKs from NXP. [The Ubuntu POC](#), [ROS2](#), and other software are enabled to work with robotics development systems.

NavQPlus OS images can include the following.

- NXP Yocto Linux 5.15 kernel
 - gstreamer
 - eIQ AI/ML tools
 - Ethernet over USB-C Gadget mode (SSH connection to laptop using USB cable)
- Ubuntu 22.04 built on top of NXP Yocto 5.15
 - ROS2 Humble enabled

Note: The configuration changes regularly and can upgrade to newer versions.

3.3.2 Open source community

While the software is used and tested, it is continuously under development and subjects to frequent changes. The software is application level (that is, ROS2), and not hardware device level (that is, gstreamer), refer to the *open source community support for that application and not directly from NXP*.

3.3.3 Yocto Linux

NXP i.MX 8M Plus Linux is a Yocto build plus desktop image layer. Refer to the NXP Git repo and NXP documentation when building from source.

Important: *Linux on NavQPlus is based on the EVK Yocto image. However, it has board-specific DTS changes made and other packages prepared for use as a robotics development tool. The NXP Yocto tree does not include these changes. Building from scratch with Yocto is a more complex process. To maintain the functionality of the NavQPlus as provided, refer to one of the third-party images/source trees.*

For details, see the i.MX Yocto Project User Guide (document: [IMXLXYOCTOUG](#)).

3.3.3.1 Other NXP EVK software

NavQPlus is a derivative of the NXP EVK. For a complete view of all the software available, refer to the NXP website [i.MX 8M Plus](#).

3.4 Experimental nature

NavQPlus is experimental and contains a new set of boards and peripherals. Therefore, expect and plan for software enablement to undergo several iterations. The intent is to provide an Ubuntu Proof of Concept (PoC) "user-friendly Linux" with packages and other tools included instead of the traditional highly optimized and stripped down Yocto-based distribution for Linux operating systems, used in embedded products. This Ubuntu PoC is built on top of Yocto. Therefore, it is possible to reduce and optimize it for full commercial deployment.

3.5 Ubuntu Proof of Concept

The software available for the NavQPlus includes an Ubuntu desktop environment based on the Ubuntu open source distribution as a Proof of Concept (PoC).

Note: *This Ubuntu PoC is NOT an official Canonical distribution.*

Canonical outlines this mission philosophy of supporting free and open software here: <https://ubuntu.com/community/ethos/mission>.

3.5.1 Canonical Ubuntu commercial support

Canonical independently supports the commercial development on NXP i.MX processors. To provide commercial support for Canonical Ubuntu on NXP processors. They are available on a contractual basis.

3.5.2 Third-party commercial support

NXP has created NavQPlus in cooperation with Emcraft, which is one of our Gold partners,

Note: *Various NXP partner program partners are available for commercial support including hardware and software support.*

For further details, refer to: <https://www.nxp.com/design/partner-marketplace:PARTNER-MARKETPLACE>.

4 Quick start guide

This section lists the steps to start using the NavQPlus in no time. For more information, refer to [Section 5 "NavQPlus interface usage"](#), [Paragraph](#), and the other parts of this GitBook.

4.1 Download an Ubuntu image

Download the latest Ubuntu image for NavQPlus in the *.wic file format from GitHub:

- For use with iRobot Create3 (AKA Turtlebot4), refer to <https://github.com/rudislabs/navqplus-create3-images/releases>.
- For use with NXP MR-B3RB, refer to <https://github.com/rudislabs/meta-navqplus-apt-ros>.

4.2 Download the Universal Update Utility tool

Universal Update Utility (UUU) is an NXP command-line tool that runs on a computer and can communicate directly with the MPU on the NavQPlus.

To [flash the eMMC memory on NavQPlus](#), you must download the uuu tool.

Download the latest release from GitHub: <https://github.com/nxp-imx/mfgtools/releases>.

Note: Make sure to download the correct application for your platform. The *.exe file is a Windows executable. The file named "uuu" without extension is a x86/64 Linux binary.

4.3 Set the boot switch to the flash mode

Before plugging in the board, find the [boot switches](#) on NavQPlus and flip them to the flash mode.

Table 1. Set the boot switch to the flash mode

Mode	Switch 1	Switch 2
SD	ON	ON
eMMC	OFF	ON
Flash	ON	OFF

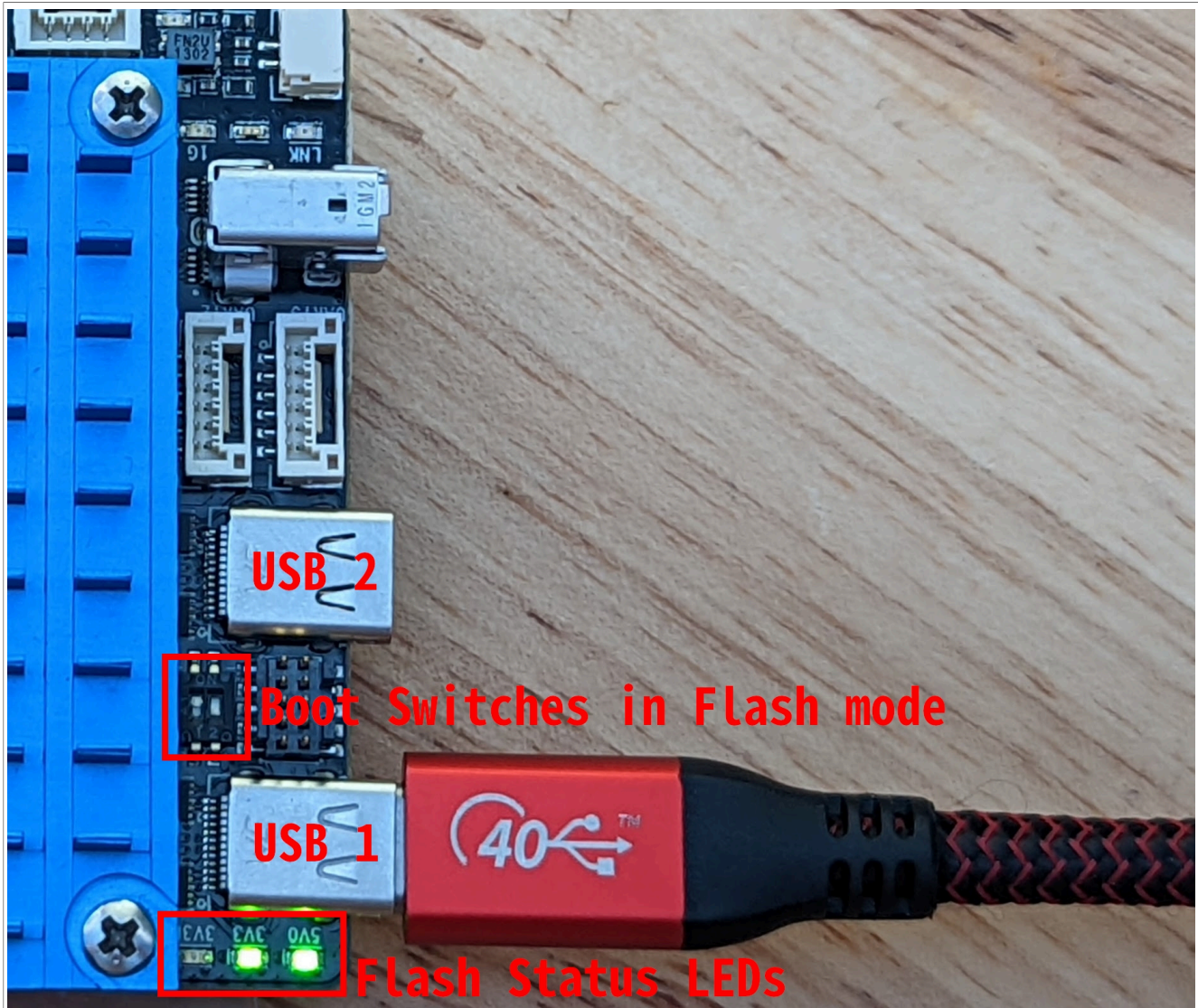


Figure 2. Set the boot switch to the flash mode

4.4 Flash the eMMC memory

Developers prefer to install the NavQPlus firmware on the eMMC memory for reliability reasons. Even though it is convenient to use the SD card for development. Especially, if you are flying a drone, vibrations can cause occasional failure of the physical connections to an SD card.

1. Connect the NavQPlus to your computer using the leftmost USB-C port (USB 1). The two status LEDs must light up as shown below.

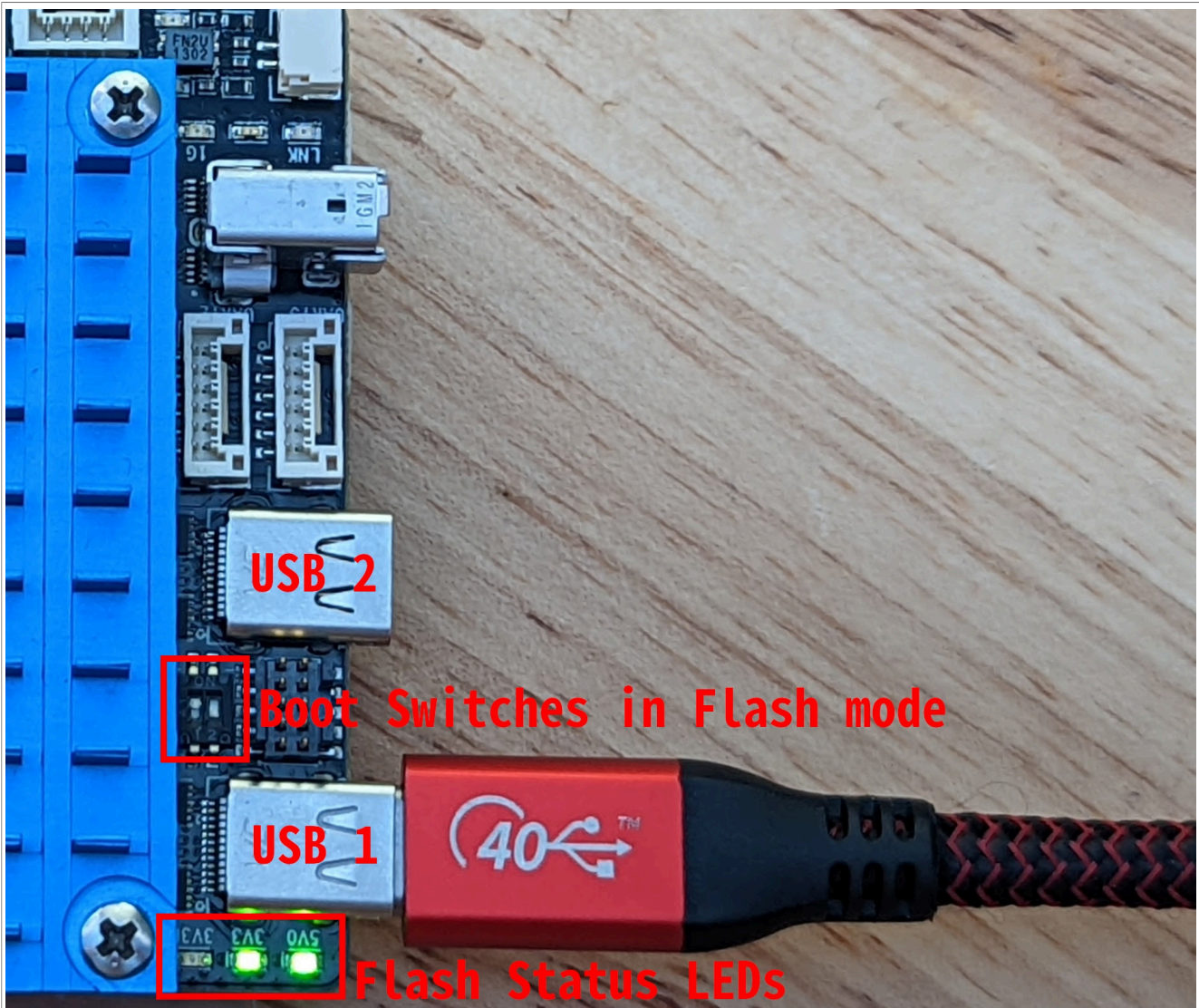


Figure 3. Set the boot switch to the flash mode

2. Open a command-line window.
3. To check that UUU recognizes the NavQPlus firmware, run the following command.
Note: Make sure that you are in the correct directory where the UUU file is available and the image file. Otherwise, you must add the path to the file.

For Linux users

```
./uuu -lsusb
```

For Windows users:

```
./uuu.exe -lsusb
```

```
./uuu -lsusb
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.4.193-0-ge56424c

Connected Known USB Devices
  Path      Chip      Pro      Vid      Pid      BcdVersion
  =====
  2:34      MX865    SDPS:    0x1FC9  0x0146  0x0002
```

Figure 4. Check that UUU recognizes NavQPlus

Ensure device detection. To flash your board, use the command below. Adapt the filename and version to match the image file that you downloaded in step 1.

For Linux users:

```
.uuu -b emmc_all navqplus-image-<version>.wic
```

For Windows users:

```
.uuu.exe -b emmc_all navqplus-image-<version>.wic
```

Once this process finishes, compare the program output to the image below to ensure a successful flash process.

```
sudo ./uuu -b emmc_all navqplus-image-<version>.bin-flash_evk navqplus-image-<version>.wic
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.4.193-0-ge56424c

Success 1   Failure 0

2:34      1/ 1 [=====100%=====] SDPS: boot -scanterm -f navqplus-image-<version>.bin-flash_evk -scanlimited 0x800000
3:14      8/ 8 [Done                ] FB: done
```

Figure 5. Compare the program output

4.5 Set the boot switch to the eMMC mode

Again, change the [boot switches](#) to boot from eMMC. (For reference, see [Section 4.3](#).)

Table 2. Set the boot switch to the eMMC mode

Mode	Switch 1	Switch 2
SD	ON	ON
eMMC	OFF	ON
Flash	ON	OFF

4.6 Log in for the first time

Power on the NavQPlus by plugging in a USB-C cable to the centermost USB-C port (USB2). Make sure to provide enough power. Recommended is a >5 W supply.

The NavQPlus boots, and you can confirm that it has fully booted by observing the LEDs on board. The three LEDs by the USB1 port and the two LEDs next to the CAN bus connectors must be on.

To log into NavQPlus for the first time, you can either use the included USB to UART adapter, Ethernet, or USB gadget mode.

4.6.1 Connect USB to UART adapter

1. Connect the included USB to the UART adapter to the UART2 port on the NavQPlus.
2. Open your favorite serial console application. For example, PuTTY for Windows users and Minicom on Linux.
3. Open a serial console and set the baud rate to 115200.
If there is no output on the screen, press **Enter** to get a log-in prompt

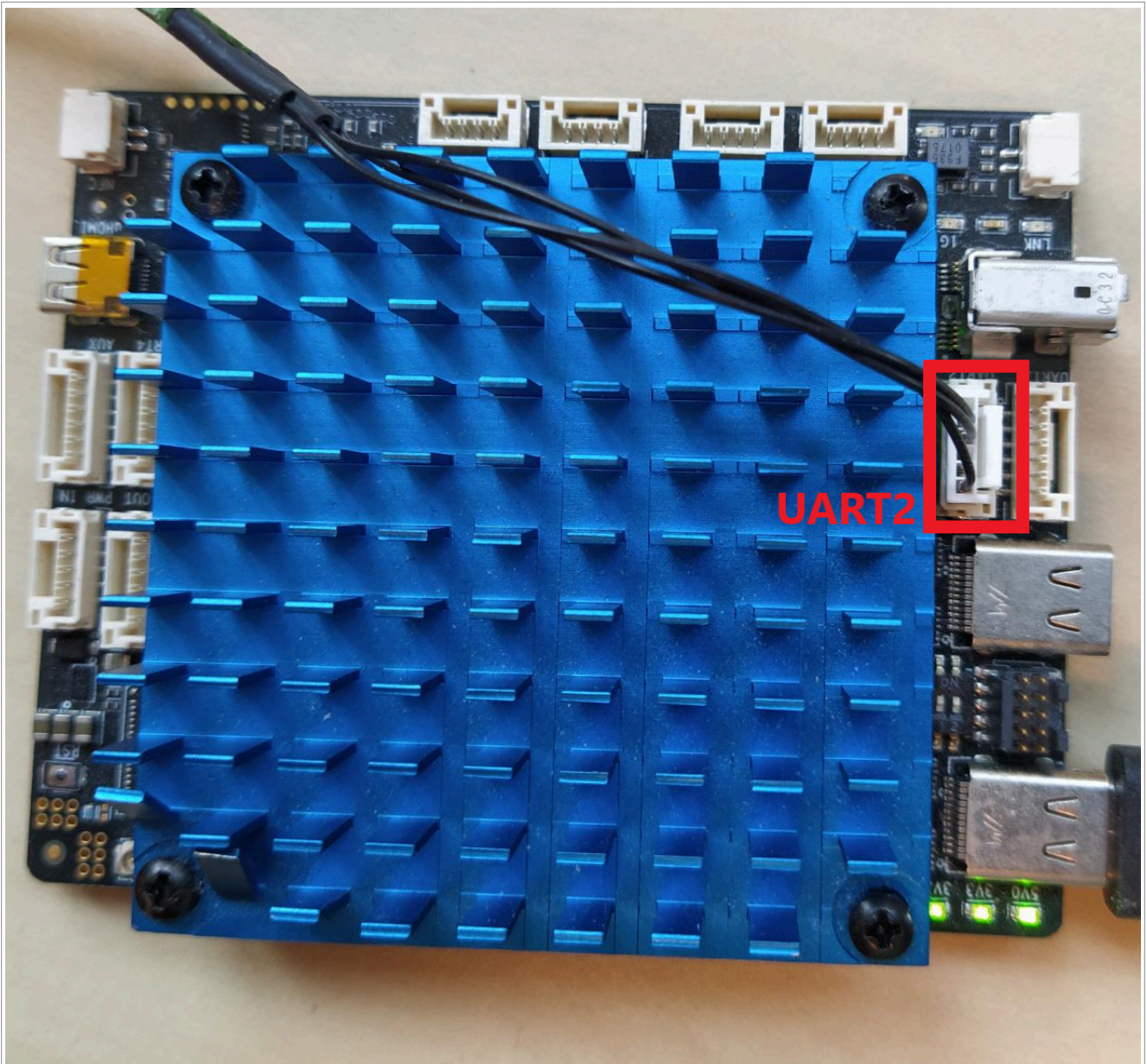


Figure 6. Connect the included USB to UART adapter

If the boot was successful, in the terminal prompts you to enter the user name and the password. The default user name/password is as follows:

```
Username: user
Password: user
```


At this point, you can start working with the NavQPlus. However, NXP recommends you to [Section "Expand image, if required"](#) and [Section "Change the default user name and password"](#).

4.6.2 Ethernet

1. To connect the board over Ethernet, connect the included ix Industrial Ethernet cable to NavQPlus.
2. Connect the RJ45 connector to your computer, switch, or router on your local network.

You can log into NavQPlus over SSH. For more information on this setup, refer to [Section 5.3 "Wired network connections"](#).

The default host name for NavQPlus is `imx8mpnavq` or use the IP address from your board instead. To SSH into NavQPlus, you can run the following command:

```
ssh user@imx8mpnavq.local
```

After a successful connection, the terminal prompts you to enter the password. As mentioned above, the default password is `user`.

4.6.3 USB gadget Ethernet

NavQPlus statically assigns the IP address `192.168.186.3` to the `usb0` network interface. To use USB gadget Ethernet to connect to NavQPlus, you must assign a static IP to your existing gadget Ethernet interface on your computer.

First go to your network settings and click the plus icon on the top right.

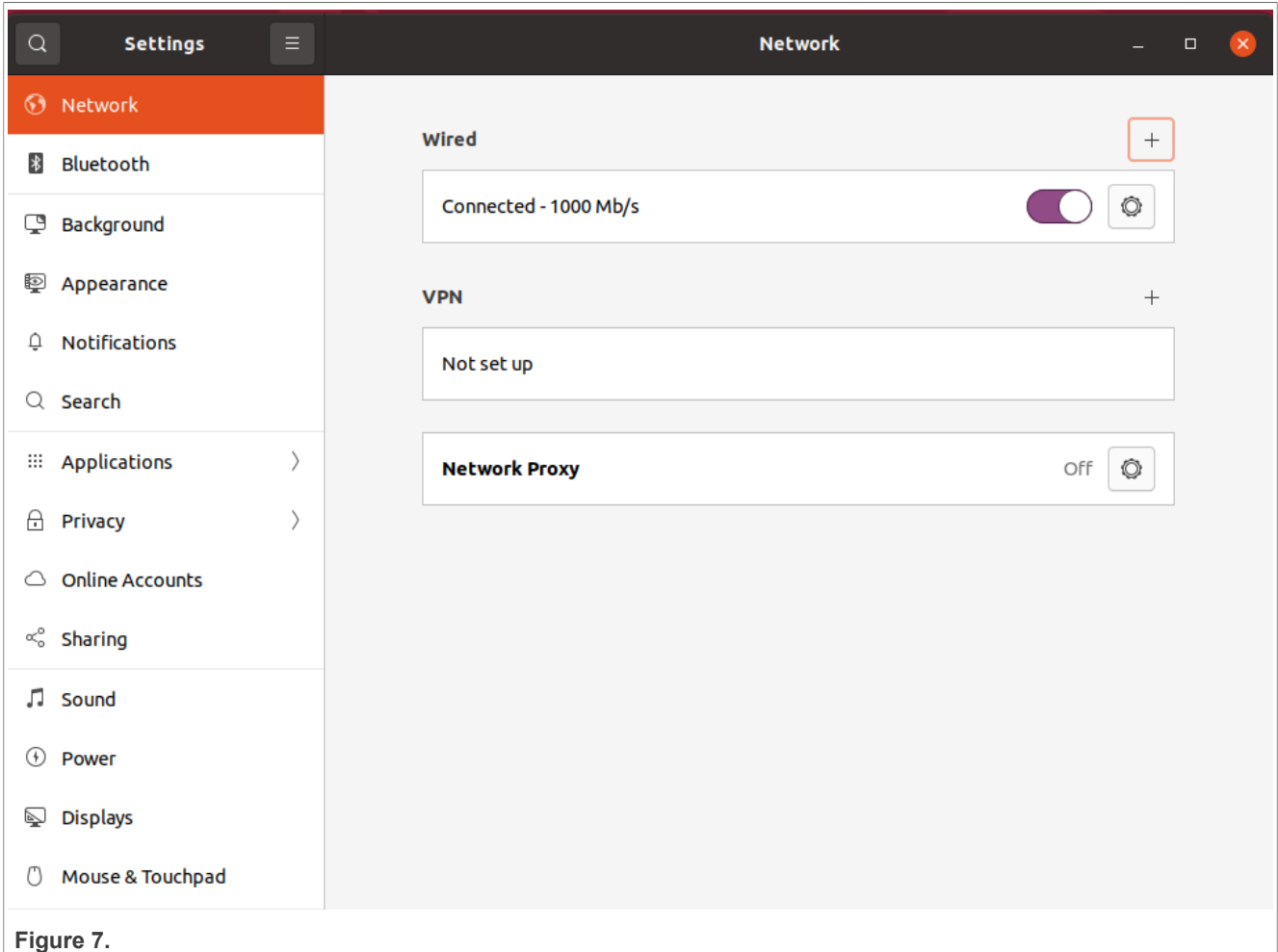


Figure 7.

The network configuration is as follows:

IP Address: 192.168.186.2

Network Mask: 255.255.255.0

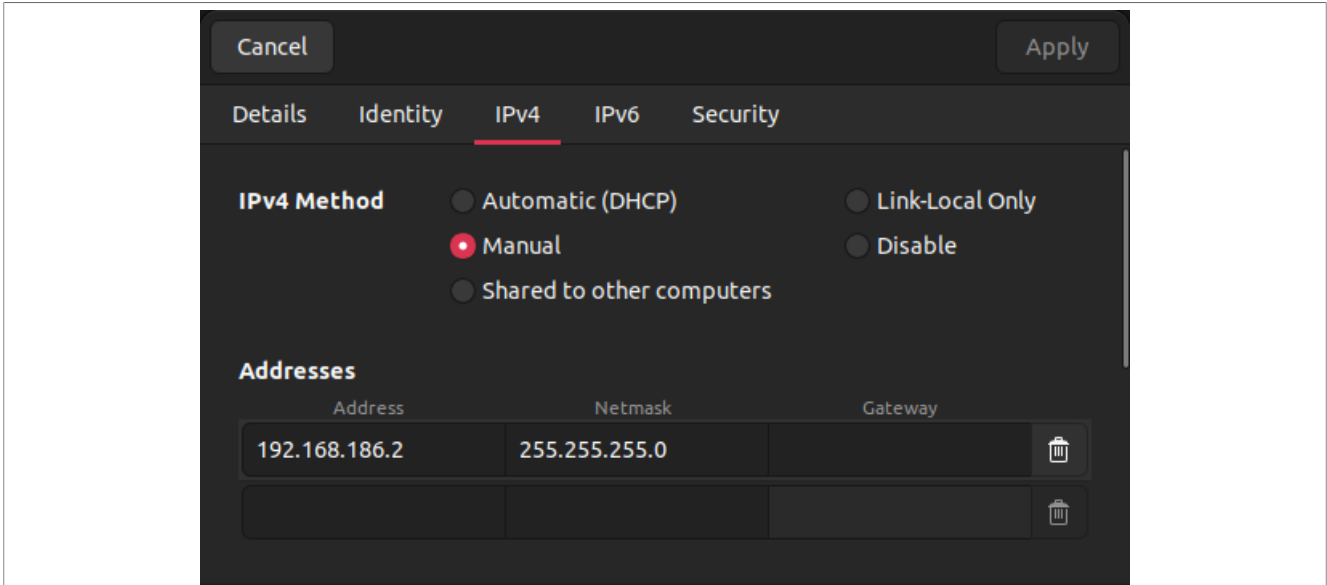


Figure 8. Network manager connection profile.

Once you have set up your USB-C gadget Ethernet interface on your computer, you can SSH by running:

```
ssh user@imx8mpnavq.local
```

4.7 Expand image, if required

Note: This step of expanding the image is not used with MR-B3RB.

To use all the available storage, you must expand the flashed images.

After logging into the NavQPlus, open a terminal and run:

- Expand the image on the eMMC memory (if you chose to flash the image on the eMMC memory):

```
echo -e "d\n2\nn\np\n2\n196608\n\n\nw" | sudo fdisk /dev/mmcblk2 && sudo  
resize2fs /dev/mmcblk2p2
```

- Expand the image on the SD card (if you chose to flash the image on the SD card):

```
echo -e "d\n2\nn\np\n2\n196608\n\n\nw" | sudo fdisk /dev/mmcblk1 && sudo  
resize2fs /dev/mmcblk1p2
```

4.8 Change the default user name and password

To change the default user name and password, use the commands below.

User name: Replace <new_username> with your desired user name:

```
usermod -l <new_username> user  
mv /home/user /home/<new_username>
```

Password: The window prompts you to enter the current and new passwords.

```
passwd
```

4.9 Next step

With the NavQPlus setup complete, you can start to install other software packages and run your own code. For example, ROS2 (Robot Operating System) is a commonly used framework for controlling robotic systems, with plenty of compatible packages being available.

5 NavQPlus interface usage

This section covers the basics of the NavQPlus connection.

5.1 Power supply options

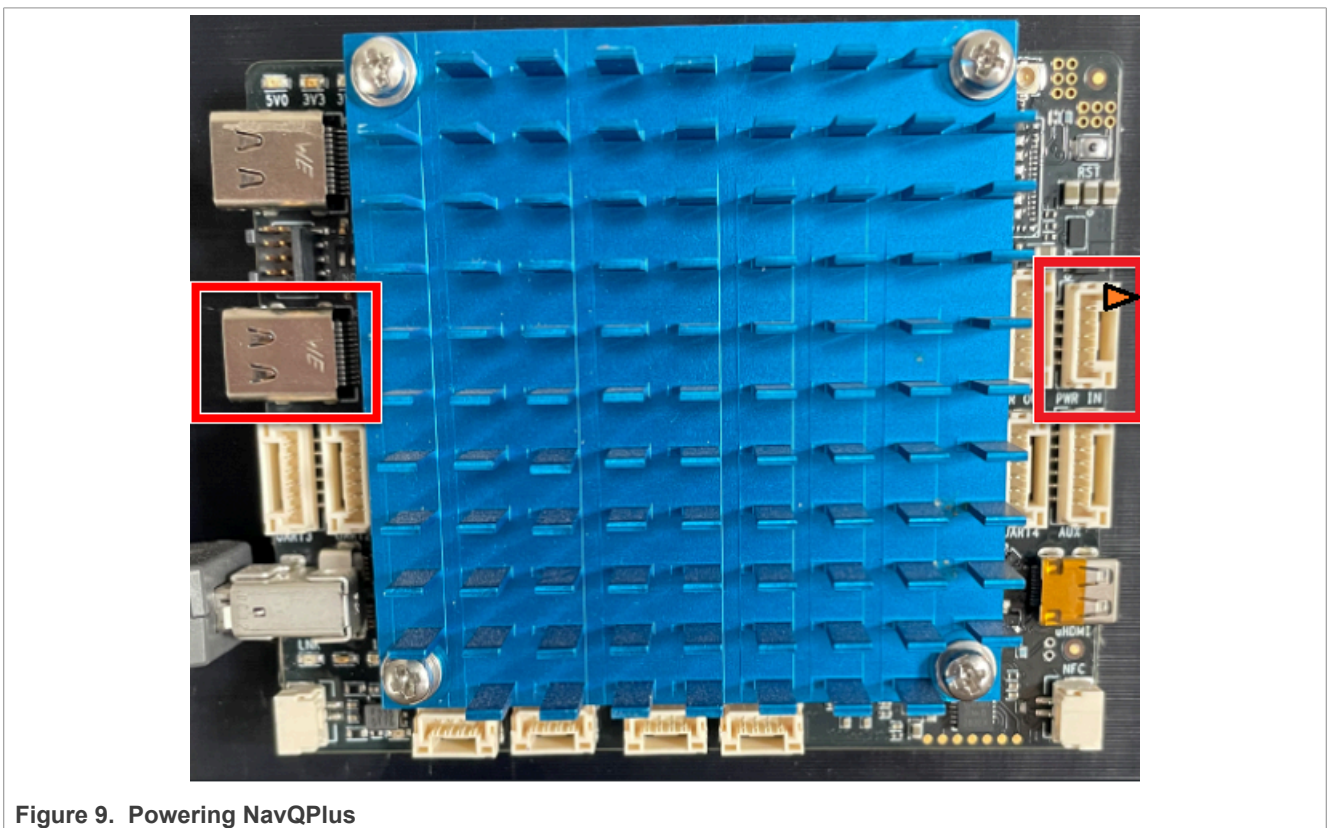


Figure 9. Powering NavQPlus

The software can configure the USB-C power management logic such that it limits current or resets the power management when both the bootloader and Linux image are running. This results in an unexpected behavior at power up. If you suspect something unusual on power up or the board reboots, power through `PWR_IN` first. The `PWR_IN` port accepts an input in the 5 V-20 V* range. However, full validation or characterization is not done. The `PWR_IN` port pinout schematic is as follows:

- The center pin is unused.
- Pins 1 and 2 are the power input.
- Pins 4 and 5 are negative (GND).

The input power voltage connection `PWR_UNREG` is 5 V-20 V.

5.1.1 Unexpected power resets

You can encounter unexpected resets due to any of the following issues.

- [Section 5.1.1.1 "Insufficient current "](#)
- [Section 5.1.1.2 "Power glitch from other boards"](#)
- [Section 5.1.1.3 "Linux Image problems"](#)

5.1.1.1 Insufficient current

Insufficient current situation occurs when the power supply is unable to provide enough current to the board or the board and the peripherals. This can result in a brown out condition when internal blocks, interfaces, and peripherals turn on and draw more current.

If you have a bench or wall adapter power supply limitation, supplying power with a Li-Po battery can help debug.

5.1.1.2 Power glitch from other boards

The `PWR_IN` port connects to an onboard power control switch that can sense reverse current spikes on the power supply and deassert a power good signal, causing a reset. This happens when another device is sharing the power supply and glitches the power supply when plugged in live. You may need to add a reverse blocking diode and more bulk capacitors on a power distribution board or inline with the power cord.

5.1.1.3 Linux Image problems

As mentioned above, you must check the specific Linux image when powering from USB-C. The check must ensure that the boot Linux kernel does not reset USB power on startup after the boot loading completes.

The bootloader configures the USB-C, which allows the Linux kernel to re-configure/reset it and switch it off again.

5.2 Serial console

The following sections provide information on connecting to the serial console.

The NavQPlus kit includes an FTDI type USB-C to UART adapter cable and a small adapter board for this cable to the serial port JST-GH connector. A serial debugging console on the NavQPlus uses this adapter.

5.2.1 USB to UART adapter

Connect the USB to UART adapter to the UART2 port on the NavQPlus. Open your favorite serial console application. For example, PuTTY for Windows users, Minicom on Linux. Open a serial console and set the baud rate to 115200. If there is no output on the screen, press **Enter** to get a log-in prompt.

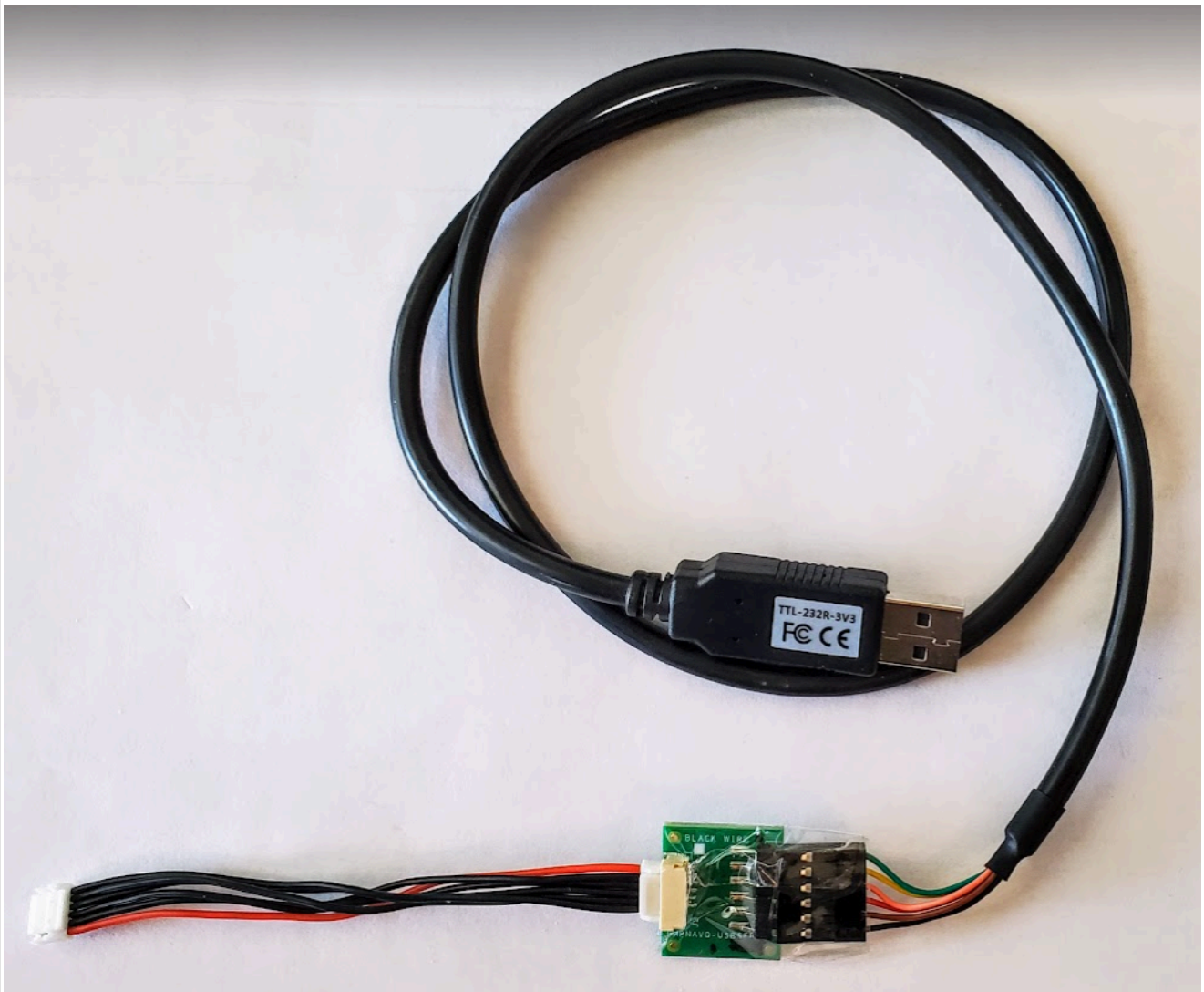


Figure 10. USB to UART adapter

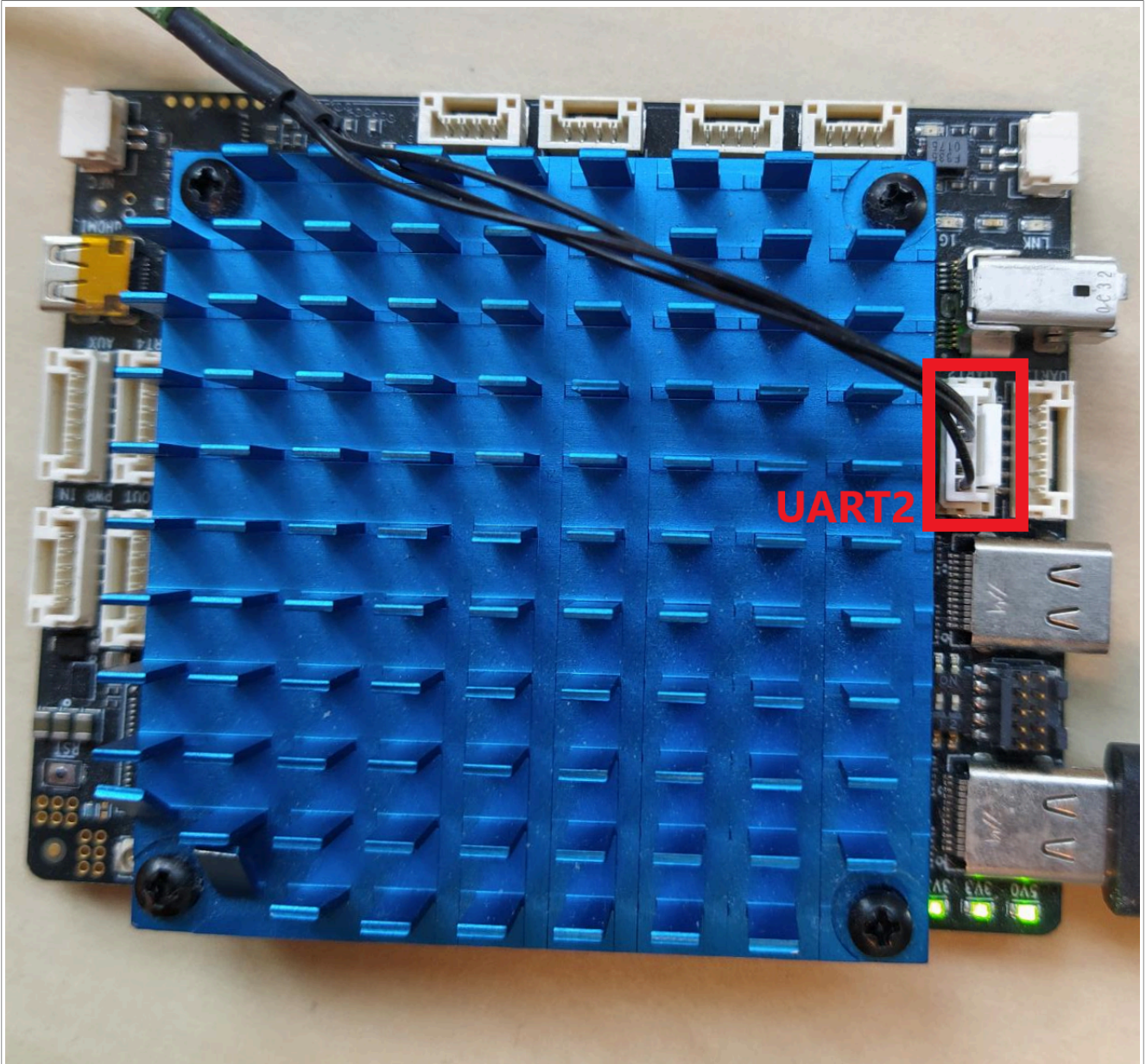


Figure 11. USB to serial cable and adapter board (Not exactly as shown)

Connect a USB cable+ adapter to your computer. Plug in the JST-GH connector from the adapter into the UART2 (A53 Debug/Console) port on NavQPlus.

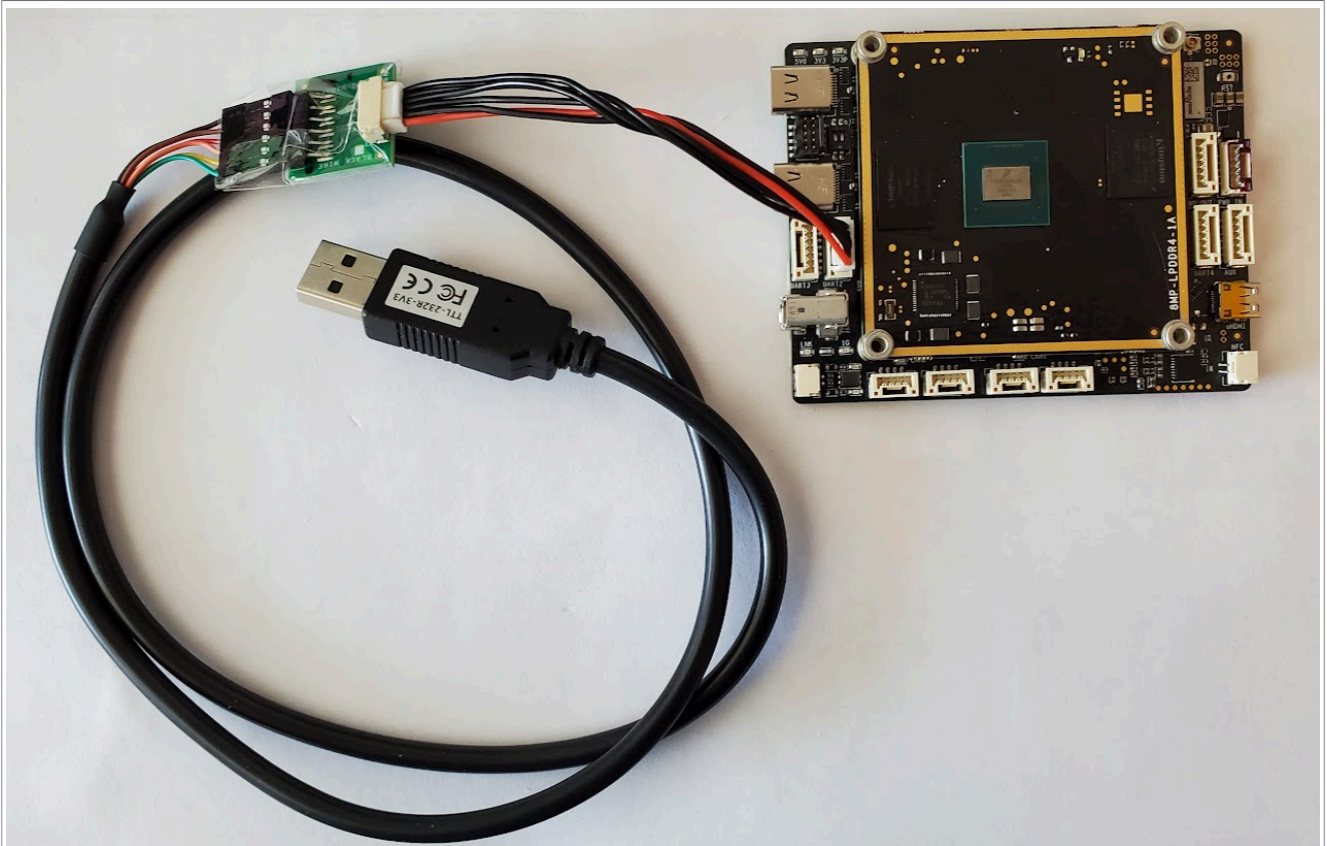


Figure 12. Plug in the JST-GH connector from the adapter

5.2.2 Serial terminal software

Use your favorite serial console software such as PuTTY (Minicom, MobaXTerm, or screen) to access the NavQPlus serial console directly.

5.2.2.1 Console baud rate

The baud rate is 115200.

Note: You can use the serial console to observe the full boot sequence including U-Boot. The terminal program on your PC should not disconnect on reboot or reset of the NavQPlus. This is important because the connection to the PC is really to the USB-UART adapter board inside the TTL-232R-USB cable.

5.2.3 Successful boot

The Linux boot details appear on the terminal when the correct code loads and the boot switches connect to the correct boot source (SD card vs EMMC).

5.2.3.1 Default user name and default password

The system prompts you for the user name and then the password. The default user name/password is as follows:

```
Username: user
```



```
Password: user
```

At this point, you can start using Linux on the NavQPlus.

5.2.3.2 Other boot details

NavQPlus must boot normally after applying power. While booting, the device provides detailed information to the serial console, accessible on the [UART2 connector](#). Using the console can provide invaluable debugging information that is difficult to find elsewhere.

Once the boot process completes, the system console is available by starting an SSH session over (USB-) Ethernet or Wi-Fi.

Important: There are [DIP switches on board that select the boot source](#) as either SD card or onboard eMMC memory. Ensure that the settings match your requirements.

5.2.3.2.1 Power requirements for successful boot

If the applied voltage at `PWR_IN` is too low, the boot process could hang upon initializing the CPU in the Linux kernel. Make sure that your power input is 5 V-20 V. When all on-chip peripherals are enabled, the NavQPlus does not draw more than 4 watts of power and it has relatively low current requirements. However, the board can also supply power to external devices. Ensure that the power source is able to maintain a stable voltage level at the requested current.

Warning: . As the software progresses, some Linux kernel/boot configurations can limit the current or switch modes of the USB-C port hardware when powering from USB-C. This can result in a hang or a reboot. If you suspect this is the case, power from the `PWR_IN` port instead.

5.2.3.2.2 U-Boot - Observe the full boot process through the serial console

The boot process starts with U-Boot, loading the device trees in the bootpartition and loading the Linux kernel. If you want to observe the complete boot process, you must monitor the [serial console](#) output on [UART2](#) by using the USB to serial converter cable provided.

5.2.3.2.3 Shell login

Once the NavQPlus boots to the shell, the default login credentials are as follows.

```
Username: user \ Password: user
```

5.3 Wired network connections

The NavQPlus can connect via Ethernet in several different ways.

- Through the IX Industrial Type A port. The provided IX to RJ45 adapter cable allows connecting to RJ45. This is the preferred option.
- Using the USB-C port with "gadget mode" Ethernet over USB
- Through the 100 Base T1 automotive Ethernet. A media port converter such as 100Base-Tn RJ45s RDDRONE-T1ADAPT, or a switch such as MR-T1ETH8 allows connection to aRJ45 100Base-T port.
- Using a USB-C Ethernet adapter dongle.

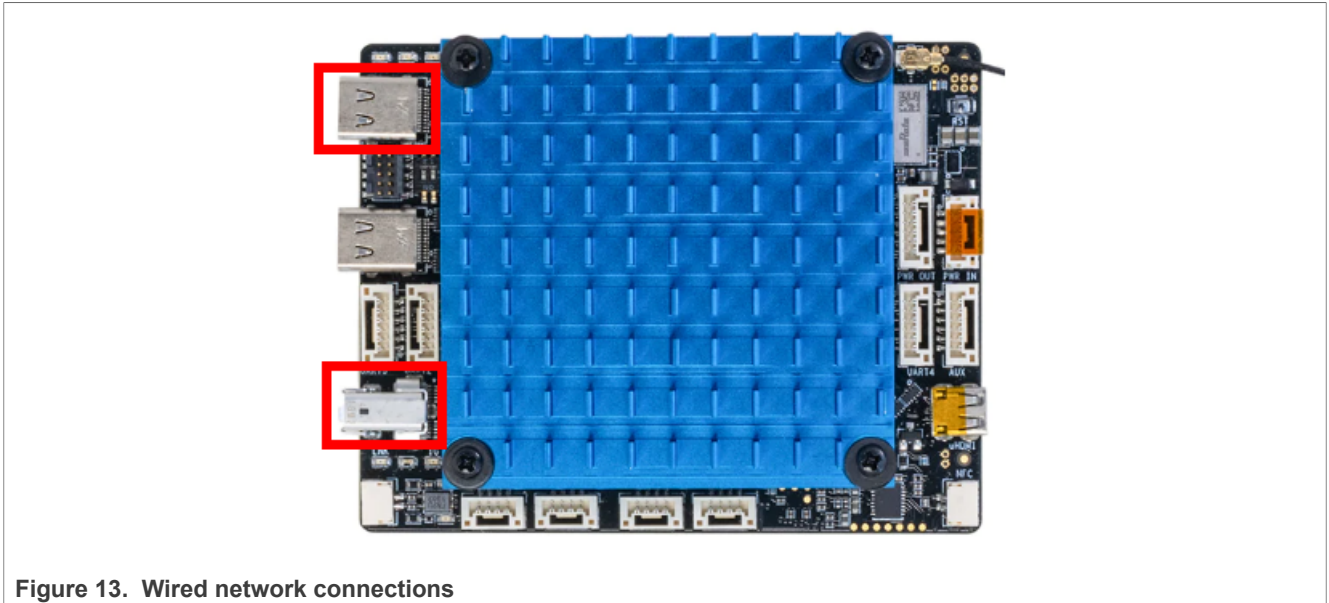


Figure 13. Wired network connections

5.3.1 USB to Ethernet adapter

The NavQ+ connects through the USB-C port, as shown in the following example.

For this method, you need a separate adapter to convert the USB-C to RJ45.

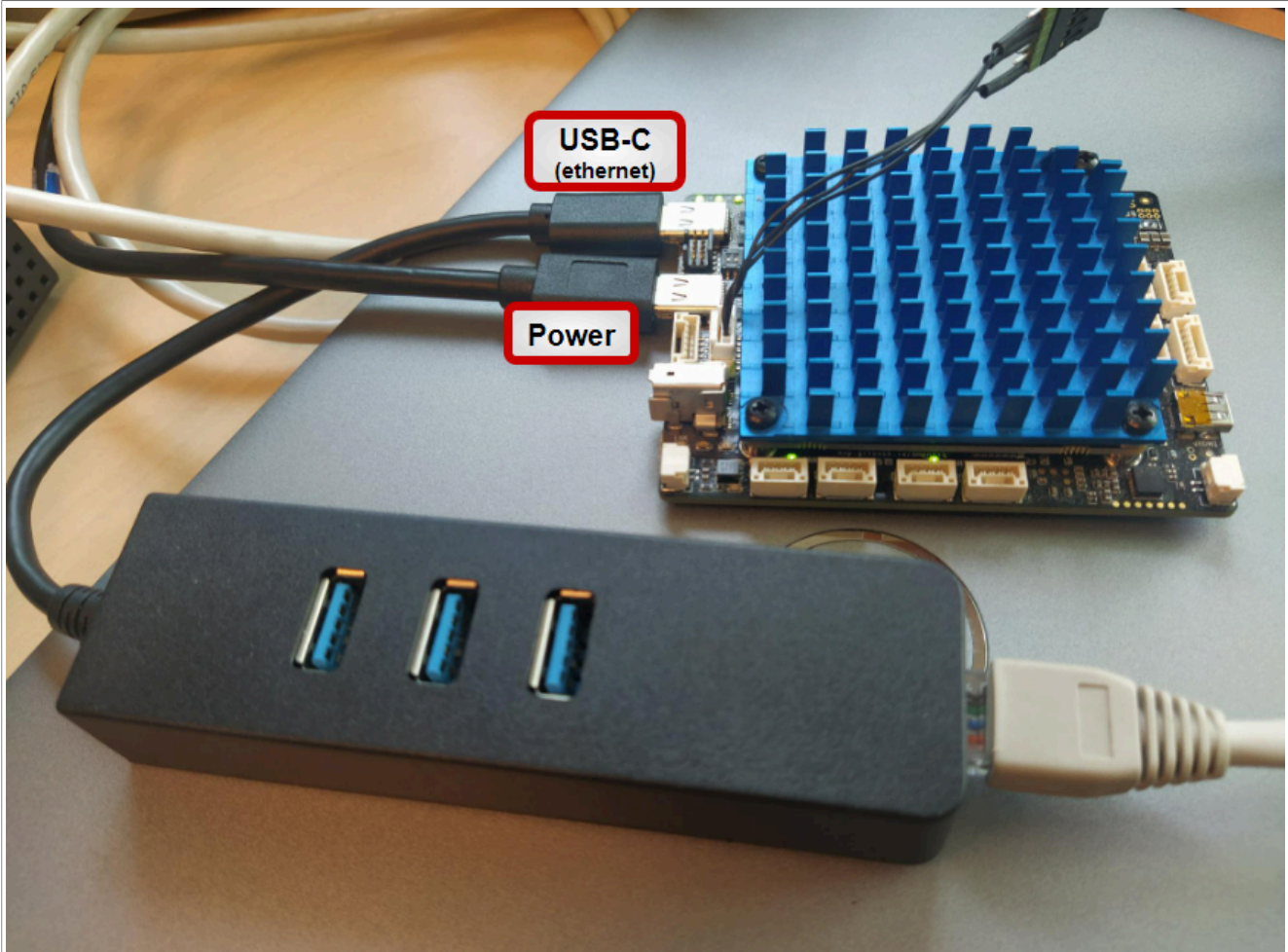


Figure 14. Example of the USB-C Ethernet connection

To connect the board through SSH, it must connect the same network using a router and a network cable. Next, connect the NavQ+ and the PC to it. The connection establishes SSH between the PC and the NavQ+.

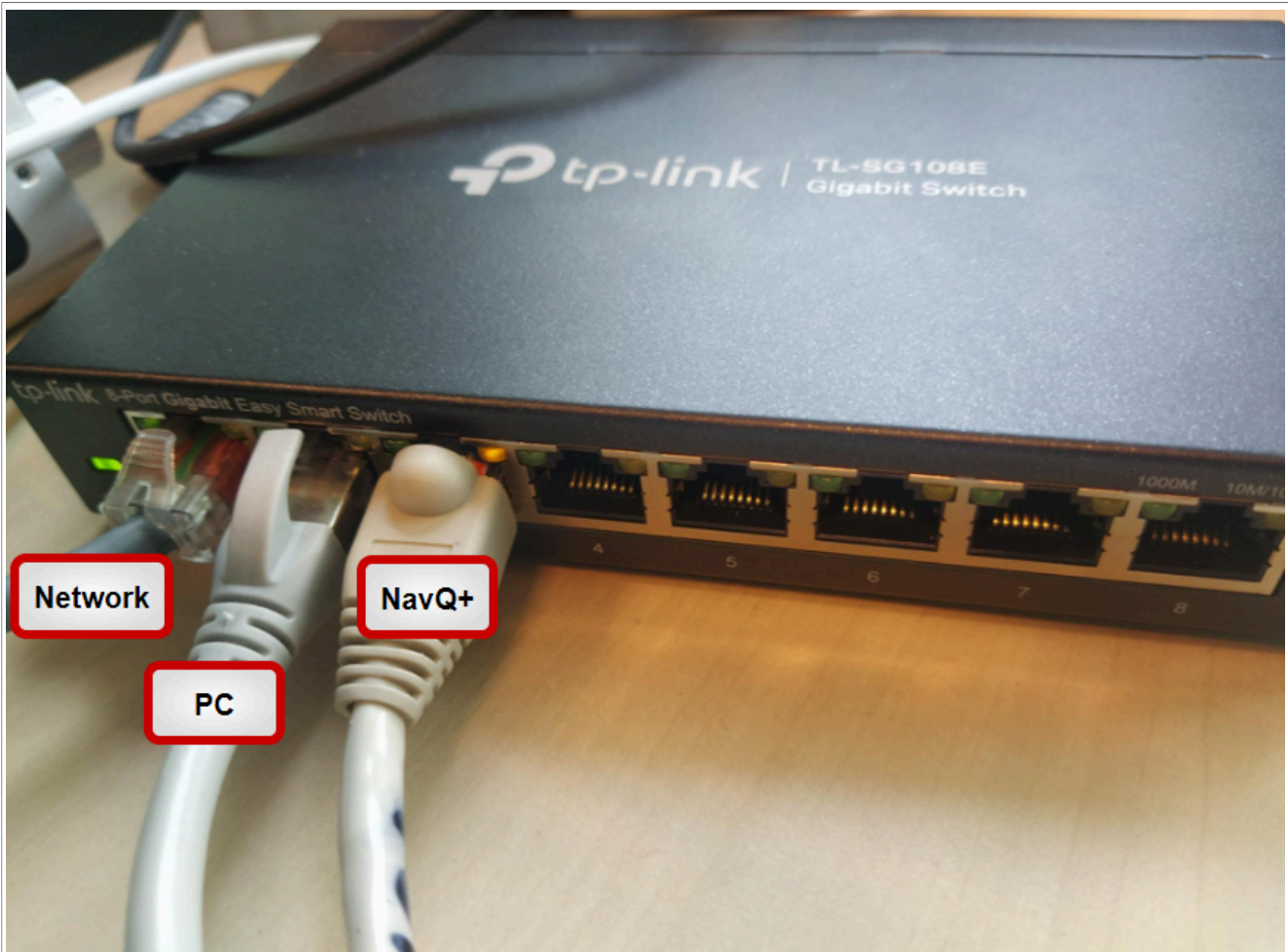


Figure 15. PC and NavQ+ are connected to the same router

Warning: Not all USB to Ethernet adapters is to Ethernet adapters supported. Sometimes, the board reboots when powering up with an adapter. While still suboptimal, unplug the Ethernet, circumvent the reboot, and let the board boot first. Once the board boots, reattach the adapter and reconnect the Ethernet.

Next, connect through SSH using the following code on the PC terminal:

```
ssh user@imx8mpnavq.local
```

Or you can input the IP address of your NavQ+:

```
ssh user@<NavQ+'s IP address>
```

The terminal prompts you to input the password.

The default password is **user**.

Important: You can also find your IP address using the command: `ifconfig`.

To connect the NavQPlus WiFi to a local network, perform the steps in the following section.

5.4 Wireless networking

This section provides information on configuring WiFi, system host name, and uername/password.

5.4.1 Configuring WiFi on NavQPlus

To connect NavQPlus to a WiFi network, use the `nmcli` command. The interface is relatively straightforward to connect with `nmcli` using the following command.

```
sudo nmcli device wifi connect <network_name> password "<password>"
```

If there is an issue in connecting to a network, see if it is visible by using the following command.

```
sudo nmcli device wifi list
```

Once NavQPlus successfully connects to a WiFi network, it reconnects to the same network even after a reboot.

5.4.2 Identify the network

To identify the WiFi network the NavQPlus is connected without using `sudo`, use the following command.

```
nmcli device wifi list
```

Or, if running with `sudo` it is the network prefixed with a star.

5.4.3 Connecting to NavQPlus over WiFi

To connect over a local WiFi network, SSH into the NavQPlus over WiFi, and run the following command.

```
ssh <username>@<hostname>.local
```

Or, depending on the network setup, run the following command.

```
ssh <username>@<hostname>
```

5.4.4 WiFi - nmtui

This section provides information on setting up the NavQ+ to connect with WiFi.

5.4.4.1 Configuring WiFi on NavQPlus <

An alternative tool to connect NavQPlus to the local Wi-Fi network is the `nmtui` command. This command presents a GUI on the terminal to help connect with the Wi-Fi. The interface is relatively straightforward. To run `nmtui`, run the following command.

```
sudo nmtui
```

To use the non-GUI way to connect the Wi-Fi or manage the network connections, use `nmcli` and run the following command. This method is preferred.

```
sudo nmcli device wifi connect <network_name> password "<password>"
```


Once NavQPlus successfully connects to a WiFi network, you can exit the application. NavQPlus reconnects to the same network even after a reboot.

5.5 Camera usage

NavQPlus models typically ships with one Omnivision camera module, which is either from Innovave or Google Coral Camera. Third parties also have other cameras that support NavQPlus. You can also add another camera to the second MIPI port.

Note: Other USB or Ethernet cameras can also supply image data to the NavQPlus.

To take an image using an attached MIPI camera module on NavQPlus, use the `gststreamer` command. For example:

```
sudo gst-launch-1.0 -v v4l2src device=/dev/video3 num-buffers=1 ! jpegenc !
filesink location=capture1.jpeg
```

5.6 Flash the new firmware

This section provides information on changing the NavQPlus firmware and booting from SD card or EMMC flash.

5.6.1 Introduction

Note: New images are released along with an associated link.

To flash the NavQPlus, there are two options.

You can either flash the eMMC chip onboard, or flash the SD card included with the kit. The eMMC runs faster but is not removable like the SD card. The SD card is easily removeable and can be programmed quickly and directly from a PC. You can choose any of the above options.

5.6.1.1 NXP images

The official source for Linux on NavQPlus will eventually be NXP Linux Factory. However, currently it is a work in progress.

Note: This NXP Linux Factory enablement is a work in progress. Meanwhile, links to images that can be downloaded and used on the NavQPlus are provided below. Building from scratch using NXP Linux Factory and Yocto requires some advanced knowledge and is not documented currently. NavQPlus is similar to the 8MPlus EVK, but with some minor changes to the memory type, and the dtb files describing the board interfaces.

The following section lists the instructions on how to flash the SD card or eMMC.

5.6.2 Third-party images

The following are third-party images that support the NavQPlus.

- For use with iRobot Create3 (AKA Turtlebot4), refer to <https://github.com/rudislabs/navqplus-create3-images/releases>.
- For use with NXP MR-B3RB, refer to <https://github.com/rudislabs/meta-navqplus-apt-ros>.

5.6.2.1 Emcraft release images

For details, refer to <https://www.emcraft.com/products/1222#releases>.

5.6.2.2 Voxelbotics release images

For details, refer to <https://staging.voxelbotics.com/releases/>.

5.6.2.3 Rudis Labs images

For details, refer to <https://github.com/rudislabs/navqplus-create3-images/releases/>.

5.6.3 Flash an image to an SD card

The NavQPlus typically comes with a 16 GB SD card or larger that you can flash with the prebuilt Ubuntu 22.04 image. Refer to the following sections for instructions on how to flash your SD card on each platform.

Note: You must have an SD card reader available on your system to perform these instructions. Low-cost USB dongles or hubs with SD card slots are available.

5.6.3.1 Flash an SD card using Windows PC

Several free programs are available to flash an SD card with an image, we use [Win32DiskImager](#).

Once you have downloaded Win32DiskImager, insert your SD card into your computer.

Open the program.

Select the `navqplus-image-{vX.X}.wic` file as your image.

Select your **SD card** under **Device**.

DANGER: Make sure that your device selection is the correct drive letter for your SD card. You do not want to erase your hard drive. Ensure to only click **Write** after double-checking the correct drive letter.

Once the flashing completes, a prompts appears with a message that the write was successful.

5.6.3.2 Flash an SD card using Linux / Mac

To flash your SD card with the image you downloaded in [Section 5.6.3.1](#), use `dd`.

To begin, open a terminal and navigate to the folder that you downloaded the `navqplus-image-{vX.X}.wic` file.

Insert your SD card, and find the device path for it. For example, `/dev/sdX` on Linux and `/dev/diskX` on Mac.

DANGER: Be very careful that you select the correct drive path when using `dd` to flash your SD card. You can confirm with the "Disks" app on Ubuntu or the "Disk Utility" app on Mac.

Once you find the device path, run the following command in your terminal to flash the SD card.

- **For Linux:**

```
sudo dd if=navqplus-image-{vX.X}.wic of=/dev/sdX bs=1M status=progress
oflag=syncs
```

- **For Mac:**

```
sudo dd if=navqplus-image-{vX.X}.wic of=/dev/diskX bs=1m status=progress
oflag=syncsw
```

Once this is done, your SD card is flashed with the image.

5.6.3.3 Set boot switches for the SD card boot

Remember to check that your boot switches are set to boot from SD.

5.6.4 Flash the eMMC

This section provides information on how to [Section 5.6.4.1](#), [Section 5.6.4.2](#), and [Section 5.6.4.3](#).

5.6.4.1 Flash eMMC using UUU

To flash the eMMC on your NavQPlus, download the [uuu](#). NXP has created the UUU tool to flash NXP boards. Make sure to download the correct application for your platform. The file titled "uuu" with no file extension is a binary file for use on x86/64 Linux.

- After downloading uuu, find the [boot switches](#) on your NavQ+ and flip them to the "Flash" mode.
- Connect NavQ+ to your computer using the centermost USB-C port.
- Run the following command to make sure that the NavQ+ is recognized by uuu:

```
./uuu[.exe] -lsusb
```

- You should see that there is a device detected. If so, you can continue flashing.
- To flash your board, use one of the commands below depending on how the image was supplied.

Note: When flashing the eMMC an additional *.bin* file is required in addition to the *.wic* file. Recently [the latest uuu](#) was upgraded so these two files can now be included in a single zip and used without uncompressing. You may be supplied the *.zip* or the two separate files.

5.6.4.2 eMMC image in *.zip format

When eMMC image is supplied in the *.zip format, use the following command.

```
sudo ./uuu navqplus-image-{vX.X}_.zip
```

5.6.4.3 eMMC image supplied as *.bin and *.wic files

When eMMC image is supplied as *.bin and *.wic files, use the following command.

```
sudo ./uuu[.exe] -b emmc_all navqplus-image-{vX.X}.bin -flash_evk navqplus-image-{vX.X}.wic
```

Note: The SDCARD image also has a *.wic* file extension, so ensure that you are using the correct file. You cannot flash this to EMMC without the corresponding *.bin* file. However, you can use the eMMC *.wic* file to program an SDCARD, it is the same image.

Once this process finishes,, configure your [boot switches](#) to boot from eMMC.

5.6.5 Use DIP switches to set the boot source

It is possible to configure NavQPlus to boot from either SD card or eMMC. It also has a flash mode that allows you to flash either the eMMC or SD card over USB-C.

Table 3. Boot switch configuration

Mode	Switch 1	Switch 2
SD	ON	ON

Table 3. Boot switch configuration...continued

Mode	Switch 1	Switch 2
eMMC	OFF	ON
Flash	ON	OFF

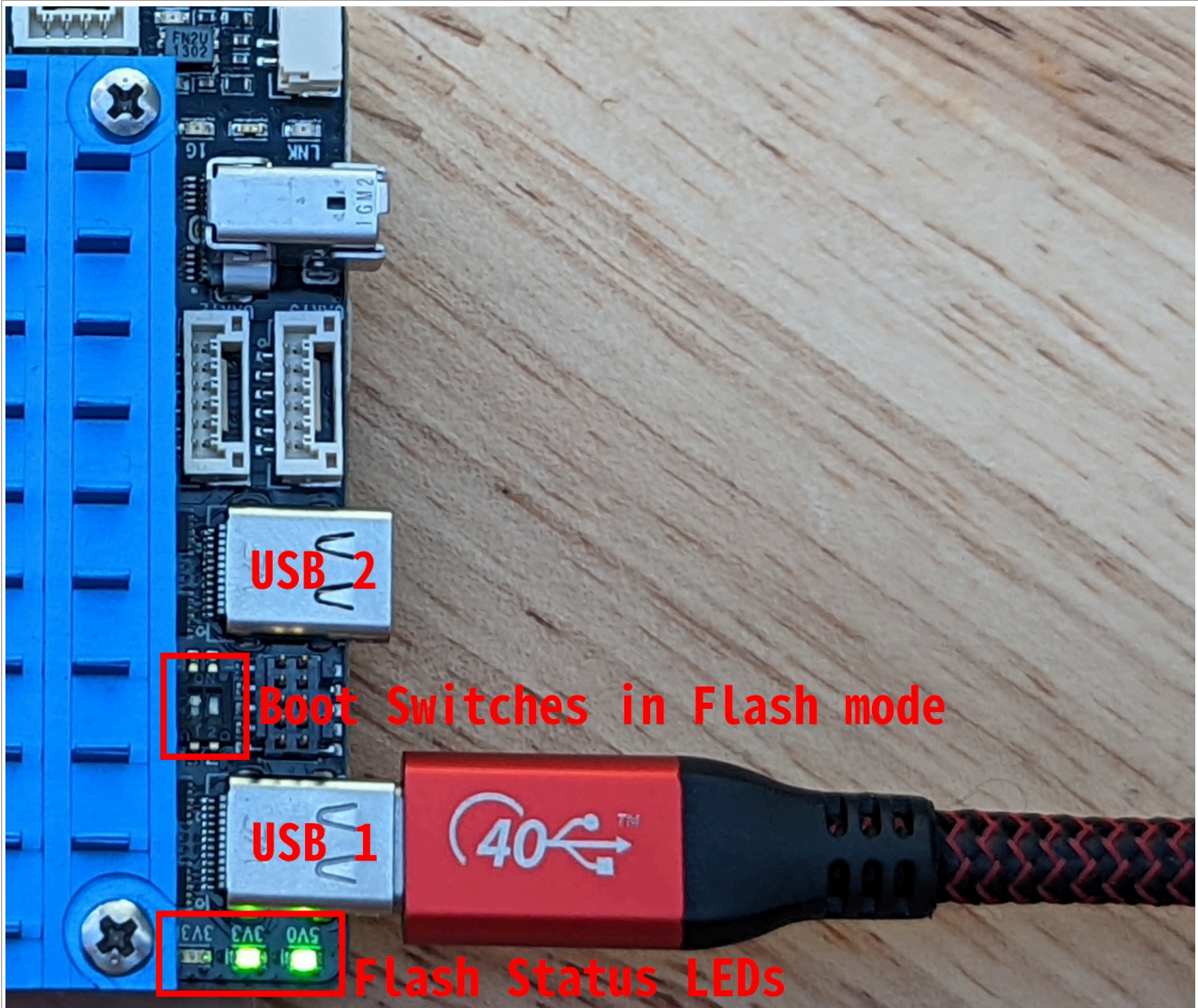


Figure 16. Set the boot switch to the flash mode

6 TurtleBot4 - iRobot Create3

NavQPlus works with the ROS2 Turtlebot4 reference platform, which is the iRobot Create3 platform. This includes Lidar interfacing.

Note: There may be more technical details, instructions, or other enablements shown on that the IRobot Create3 website, which can be relevant and repurposed for your own needs. Both documentation sources are updated regularly, so you may find it helpful to cross reference between this gitbook and the IRobot pages. For details on hardware and setup, navigate to: https://iroboteducation.github.io/create3_docs/hw/navqplus_hookup/ and https://iroboteducation.github.io/create3_docs/setup/navqplus/.

7 Simulation

This section provides information on [Section 7.1 "Examples"](#) and [Section 7.2 "Gazebo ignition"](#).

7.1 Examples

The setup and scripts provided help in configuring the environment for simulation.

7.2 Gazebo ignition

Gazebo ignition helps in modeling and simulation of vehicles using NavQPlus running ROS and other software. There are several NXP models listed for the Ignition Fuel library.

You can reference these models "live" from within Gazebo-Ignition, or download them in advance from <https://app.gazebosim.org/search;q=NXP>.

8 Note about the source code provided here in this document

The example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

9 Revision history

Table 4. Revision history

Document ID	Release date	Description
UG10110 v.1.0	14 May 2024	Initial version

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.

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.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

eIQ — is a trademark of NXP B.V.

HOVERGAMES — is a trademark of NXP B.V.

i.MX — is a trademark of NXP B.V.

PyTorch, the PyTorch logo and any related marks — are trademarks of The Linux Foundation.

Tables

Tab. 1.	Set the boot switch to the flash mode 8	Tab. 3.	Boot switch configuration 28
Tab. 2.	Set the boot switch to the eMMC mode 11	Tab. 4.	Revision history 30

Figures

Fig. 1.	NavQPlus block diagram2	Fig. 10.	USB to UART adapter18
Fig. 2.	Set the boot switch to the flash mode 9	Fig. 11.	USB to serial cable and adapter board (Not exactly as shown) 19
Fig. 3.	Set the boot switch to the flash mode 10	Fig. 12.	Plug in the JST-GH connector from the adapter 20
Fig. 4.	Check that UUU recognizes NavQPlus 11	Fig. 13.	Wired network connections 22
Fig. 5.	Compare the program output 11	Fig. 14.	Example of the USB-C Ethernet connection23
Fig. 6.	Connect the included USB to UART adapter 12	Fig. 15.	PC and NavQ+ are connected to the same router 24
Fig. 7. 14	Fig. 16.	Set the boot switch to the flash mode 29
Fig. 8.	Network manager connection profile. 15		
Fig. 9.	Powering NavQPlus 16		

Contents

1	Introduction	2	5.4.2	Identify the network	25
1.1	Features	3	5.4.3	Connecting to NavQPlus over WiFi	25
1.2	Applications	3	5.4.4	WiFi - nmtui	25
1.3	Software	3	5.4.4.1	Configuring WiFi on NavQPlus <	25
2	Caution and disclaimers	4	5.5	Camera usage	26
2.1	Resources	4	5.6	Flash the new firmware	26
2.2	Federal Communication Commission and Industry Canada information	4	5.6.1	Introduction	26
2.2.1	FCC information	4	5.6.1.1	NXP images	26
2.2.2	Canadian IC information	5	5.6.2	Third-party images	26
3	Prerequisites	5	5.6.2.1	Emcraft release images	26
3.1	Ubuntu desktop	6	5.6.2.2	Voxelbotics release images	27
3.2	Supported hardware	6	5.6.2.3	Rudis Labs images	27
3.3	Supported software	6	5.6.3	Flash an image to an SD card	27
3.3.1	NavQPlus enablement	6	5.6.3.1	Flash an SD card using Windows PC	27
3.3.2	Open source community	6	5.6.3.2	Flash an SD card using Linux / Mac	27
3.3.3	Yocto Linux	7	5.6.3.3	Set boot switches for the SD card boot	28
3.3.3.1	Other NXP EVK software	7	5.6.4	Flash the eMMC	28
3.4	Experimental nature	7	5.6.4.1	Flash eMMC using UUU	28
3.5	Ubuntu Proof of Concept	7	5.6.4.2	eMMC image in *.zip format	28
3.5.1	Canonical Ubuntu commercial support	7	5.6.4.3	eMMC image supplied as *.bin and *.wic files	28
3.5.2	Third-party commercial support	7	5.6.5	Use DIP switches to set the boot source	28
4	Quick start guide	7	6	TurtleBot4 - iRobot Create3	29
4.1	Download an Ubuntu image	8	7	Simulation	30
4.2	Download the Universal Update Utility tool	8	7.1	Examples	30
4.3	Set the boot switch to the flash mode	8	7.2	Gazebo ignition	30
4.4	Flash the eMMC memory	9	8	Note about the source code provided here in this document	30
4.5	Set the boot switch to the eMMC mode	11	9	Revision history	30
4.6	Log in for the first time	11		Legal information	31
4.6.1	Connect USB to UART adapter	12			
4.6.2	Ethernet	13			
4.6.3	USB gadget Ethernet	13			
4.7	Expand image, if required	15			
4.8	Change the default user name and password	15			
4.9	Next step	16			
5	NavQPlus interface usage	16			
5.1	Power supply options	16			
5.1.1	Unexpected power resets	16			
5.1.1.1	Insufficient current	17			
5.1.1.2	Power glitch from other boards	17			
5.1.1.3	Linux Image problems	17			
5.2	Serial console	17			
5.2.1	USB to UART adapter	17			
5.2.2	Serial terminal software	20			
5.2.2.1	Console baud rate	20			
5.2.3	Successful boot	20			
5.2.3.1	Default user name and default password	20			
5.2.3.2	Other boot details	21			
5.3	Wired network connections	21			
5.3.1	USB to Ethernet adapter	22			
5.4	Wireless networking	25			
5.4.1	Configuring WiFi on NavQPlus	25			

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.