NXP Semiconductors
User Guide

Document identifier: MCUXSDKUSBPDUG
Rev. 6, 10 July 2021

MCUXpresso SDK USB Type-C Power Delivery Stack User Guide

Contents

Chapter 1
Introduction

Many devices obtain their power from USB ports connected in laptops, cars, or wall sockets. Because of this, users need USB to fill their requirements not only in terms of data, but also to provide power or charge their devices.

The USB Power Delivery (PD) Specification enables maximum functionality over a single cable. Some specification features include an increased power level from existing USB standards, having power direction no longer fixed, optimized power management across multiple peripherals, intelligent and flexible system level management of power, and the allowed ability of low power cases.

This document includes step-by-step instructions to run the MCUXpresso USB Type-C Power Delivery stack.

Chapter 2
Software

The software is based on MCUXpresso SDK.

2.1 Folder structure

The folder structure is shown below.

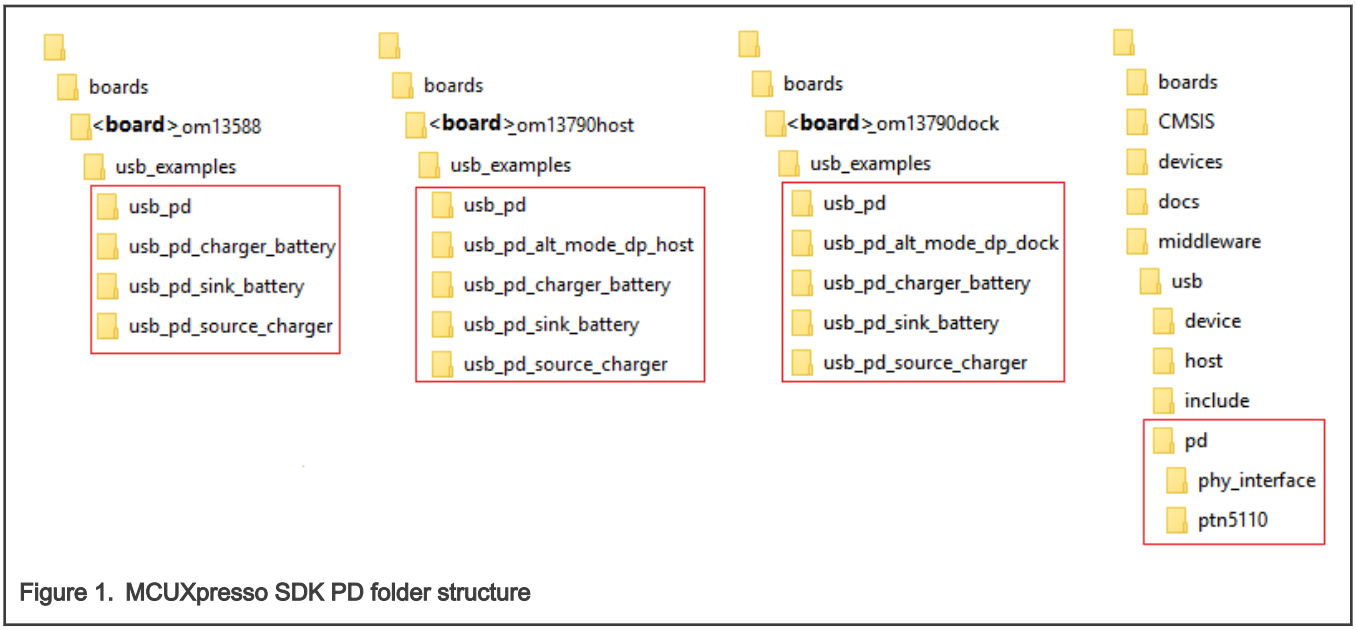

Figure 1. MCUXpresso SDK PD folder structure

The following table provides information regarding the structure and description.

Table 1. MCUXpresso SDK folder

Folder	Description
boards/ *CMSIS/* *devices/* *docs/* *middleware/* *rtos/* *tools/*	MCUXpresso SDK directories.
boards/<board>/usb_examples/	USB Type-C Power Delivery demos' projects.
middleware/usb/pd	USB Type-C Power Delivery source code.

NOTE

See the detailed description about the API of the MCUXpresso USB Type-C PD stack in *MCUXpresso SDK USB Type-C PD Stack Reference Manual* (document MCUXUSBPDAPIRM) in *docs/usb* folder.

2.2 Features

- Six examples are provided:
  - usb_pd: This demo presents all USB Type-C PD stack functions. Customers can use this demo to understand all the USB Type-C PD stack interfaces and work flows.
  - usb_pd_charger_battery: This demo simulates products that work with a battery and can charge other devices (for example, a laptop).
  - usb_pd_sink_battery: This demo simulates products that work with a battery (for example, phone).
  - usb_pd_source_charger: This demo simulates the charger product.
  - usb_pd_alt_mode_dp_dock: This demo implements the DisplayPort dock alternate mode.
  - usb_pd_alt_mode_dp_host: This demo implements the DisplayPort host alternate mode.
- The usb_pd demo supports the 5 catalogs in the USB PD3.0 compliance test (consumer/provider, provider/consumer, DRP, DRP with Try.SNK, and DRP with Try.SRC) with Ellisys EX350. The compliance tests are passed with five known issues. See section PD compliance test for more information
- The usb_pd_alt_mode_dp_host and usb_pd_alt_mode_dp_dock demos support USB PD3.0 compliance test with Ellisys EX350.
- Full toolchains are supported.
- The *MCUXpresso SDK USB Type-C PD Stack Reference Manual* (document MCUXUSBPDAPIRM) is located in the *<SDK_ROOT>/docs/usb* folder.

2.3 Building the demo

The demo projects are located in the paths below:

- *<root>/boards/<board>_<shield board>/usb_examples/usb_pd*
- *<root>/boards/<board>_<shield board>/usb_examples/usb_pd_charger_battery*
- *<root>/boards/<board>_<shield board>/usb_examples/usb_pd_sink_battery*
- *<root>/boards/<board>_<shield board>/usb_examples/usb_pd_source_charger*
- *<root>/boards/<board>_<shield board>/usb_examples/usb_pd_alt_mode_dp_host*
- *<root>/boards/<board>_<shield board>/usb_examples/usb_pd_alt_mode_dp_dock*

To build the projects, see Section 3 in *Getting Started with MCUXpresso SDK User's Guide* (document MCUXSDKGSUG) at *root/docs/Getting Started with MCUXpresso SDK.pdf*. See section Supported boards list for more information.

NOTE

1. The <shield board> is om13588, om13790host, or om13790dock. Only the om13790host supports DisplayPort host alternate mode. Only the om13790dock supports DisplayPort dock alternate mode.
2. This document introduces the common PD functions based on the usb_pd demo in <root>/boards/<board>_<shield board>/usb_examples/usb_pd.
3. For the usb_pd_charger_battery, usb_pd_sink_battery, the usb_pd_source_charger, usb_pd_alt_mode_dp_host, and usb_pd_alt_mode_dp_dock example usage, see the readme in the demos directory.
4. All USB pd example debug version for FRDM-KL27Z cannot build successfully on MCUXpresso IDE because of flash code size limitation.
5. Some examples cannot build successfully because of memory size limitation when enable PD 3.0(#define PD_CONFIG_REVISION (PD_SPEC_REVISION_30)). For details, see the readme documentation in the corresponding demos directory.
6. If one or more USB PD projects are imported in MCUXpresso IDE, the SDK debug console "UART" radio button in the main wizard page must be selected to avoid any build failures. See the following figure.

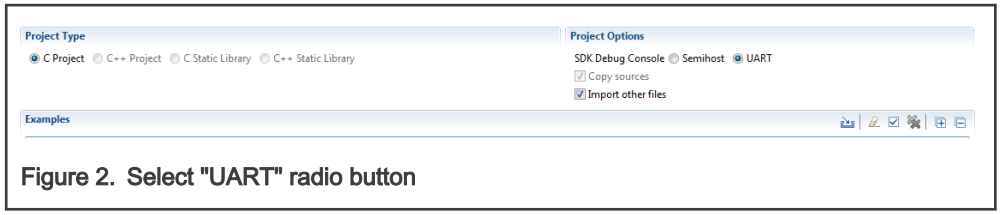

Figure 2. Select "UART" radio button

2.4 Running the demo

See Section 3 in *Getting Started with MCUXpresso SDK User's Guide* (document MCUXSDKGSUG) at *root/docs/Getting Started with MCUXpresso SDK.pdf*.

Chapter 3
Hardware

Because USB Type-C PD stack needs one provider and one consumer, the demo hardware needs two of the same set of devices; a development board and shield board, shield host board or shield dock board. For example, the LPCXpresso54114 and USB-PD/Type C shield board.

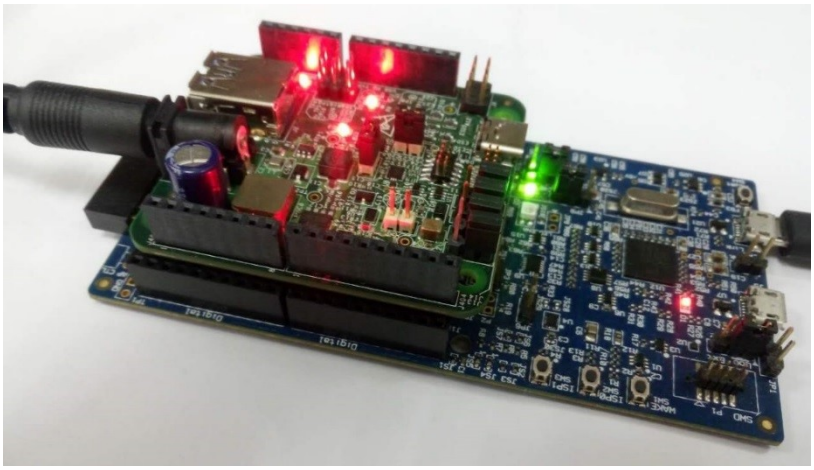

Figure 3. Hardware example - LPCXpresso54114 and shield board om13588

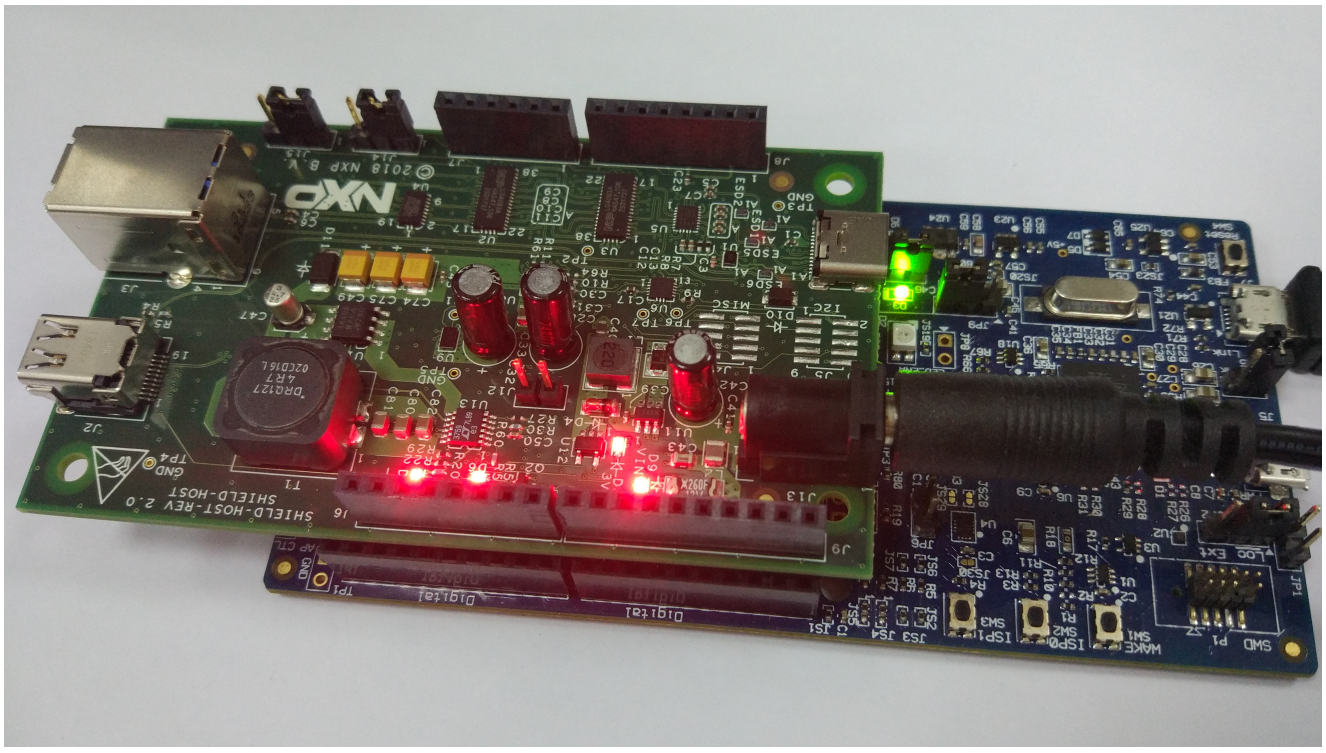

Figure 4. Hardware example - LPCXpresso54114 and shield host board om13790host

3.1 Supported boards list

- FRDM-K22F Rev A
- FRDM-K64F Rev C
- FRDM-KL27Z Rev A
- FRDM-KL28Z Rev A
- FRDM-KL32L2A4S Rev A
- IMXRT1050-EVKB
- LPCXpresso54018 Rev D
- LPCXpresso54114 Rev A
- LPCXpresso54608 Rev B
- LPCXpresso55S16 Rev A
- LPCXpresso55S28 Rev A1
- LPCXpresso55S69 Rev 1
- MIMXRT1015-EVK Rev A
- MIMXRT1020-EVK
- MIMXRT1050-EVK
- MIMXRT1060-EVK
- MIMXRT1064-EVK
- MIMXRT1170-EVK Rev B
- MIMXRT685-EVK Rev E
- MC56F83000-EVK
- MIMXRT1060-EVKB
- MIMXRT1160-EVK
- MIMXRT595-EVK

3.2 Hardware re-work

The following table provides information regarding I^2C selection.

Table 2. I^2C selection

	I2C_0	I2C_1
FRDM-K22F		√
FRDM-K64F	√	
FRDM-KL27Z	√	
FRDM-KL28Z		√
FRDM-KL32L2A4S		√
IMXRT1050-EVKB		√
LPCXpresso54018	√	

Table continues on the next page...

Table 2. I²C selection (continued)

	I2C_0	I2C_1
LPCXpresso54114	√	
LPCXpresso54608	√	
LPCXpresso55S16	√	
LPCXpresso55S28	√	
LPCXpresso55S69	√	
MIMXRT1015-EVK	√	
MIMXRT1020-EVK	√	
MIMXRT1050-EVK		√
MIMXRT1060-EVK	√	
MIMXRT1064-EVK	√	
MIMXRT1170-EVK	√	
MIMXRT685-EVK	√	
MC56F83000-EVK	√	
MIMXRT1060-EVKB	√	
MIMXRT1160-EVK	√	
MIMXRT595-EVK	√	

NOTE

- For om13588, I2C_0 means placing 1-2 for jumper J11 and J12.
- For om13588, I2C_1 means placing 2-3 for jumper J11 and J12.
- For om13790host and om13790dock, I2C_0 means placing 1-2 for jumper J14 and J15.
- For om13790host and om13790dock, I2C_1 means placing 2-3 for jumper J14 and J15.

LCPXpresso55S28: None

The connection state is as follows:

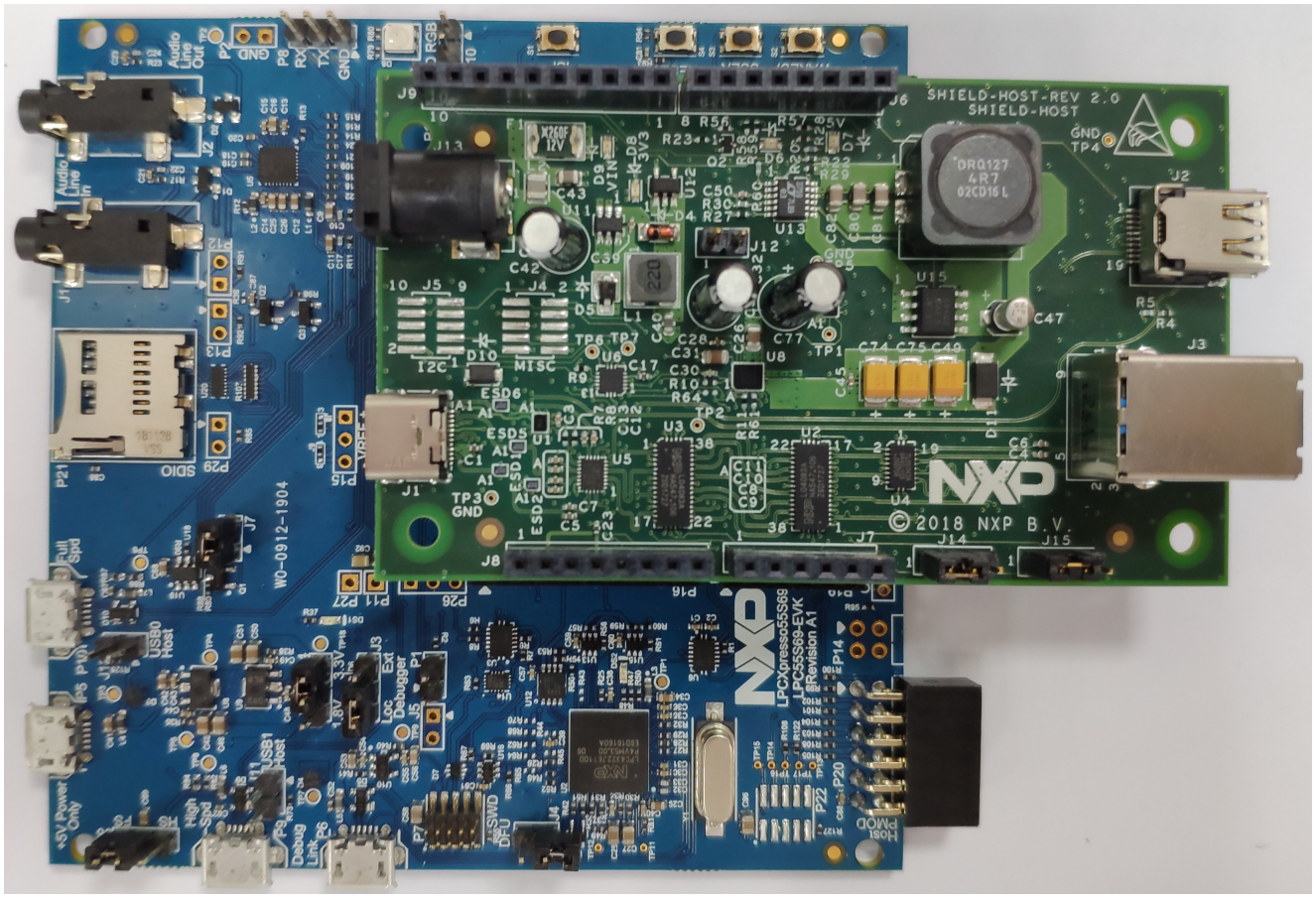

Figure 5. LPCXpresso55S28 connect shield board

LPCXpresso54114: None

The connection state is as follows:

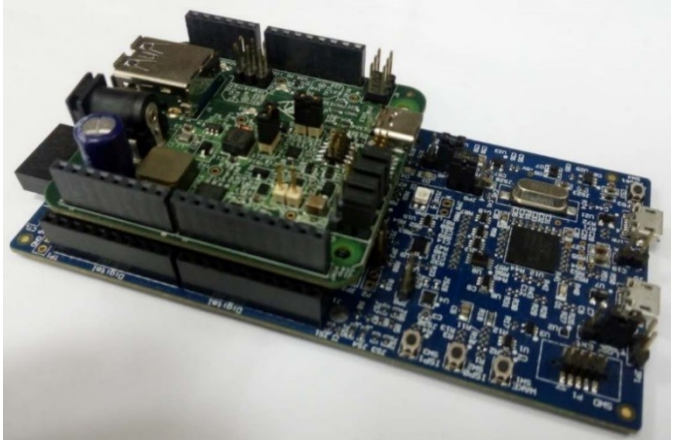

Figure 6. LPCXpresso54114 connect shield board

LPCXpresso54018 and LPCXpresso54608:

Connect the USBPD-C-SHIELD board to the Arduino® headers in the LPCXpresso54608 evaluation board.

The PTN5110 interrupt signal (nALERT) is connected to the Arduino D4 pin receptacle. The software architecture requires nALERT to be routed to an interrupt enabled IO. The LPCXpresso54608 D4 input pin PIO4_7 does not have interrupt functionality.

To work around this, use a blue wire connection between J404-1 and J403-8 for shield board om13588, between J9-1 and J6-8 for shield host board om13790host to use the PIO1_22 interrupt capable input.

The connection state is as follows:

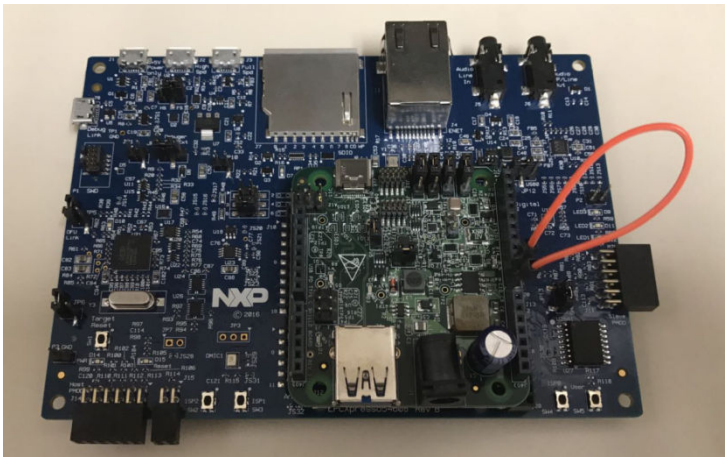

Figure 7. LPCXpresso54608 connect shield board

LPCXpresso55S69: None

The connection state is as follows:

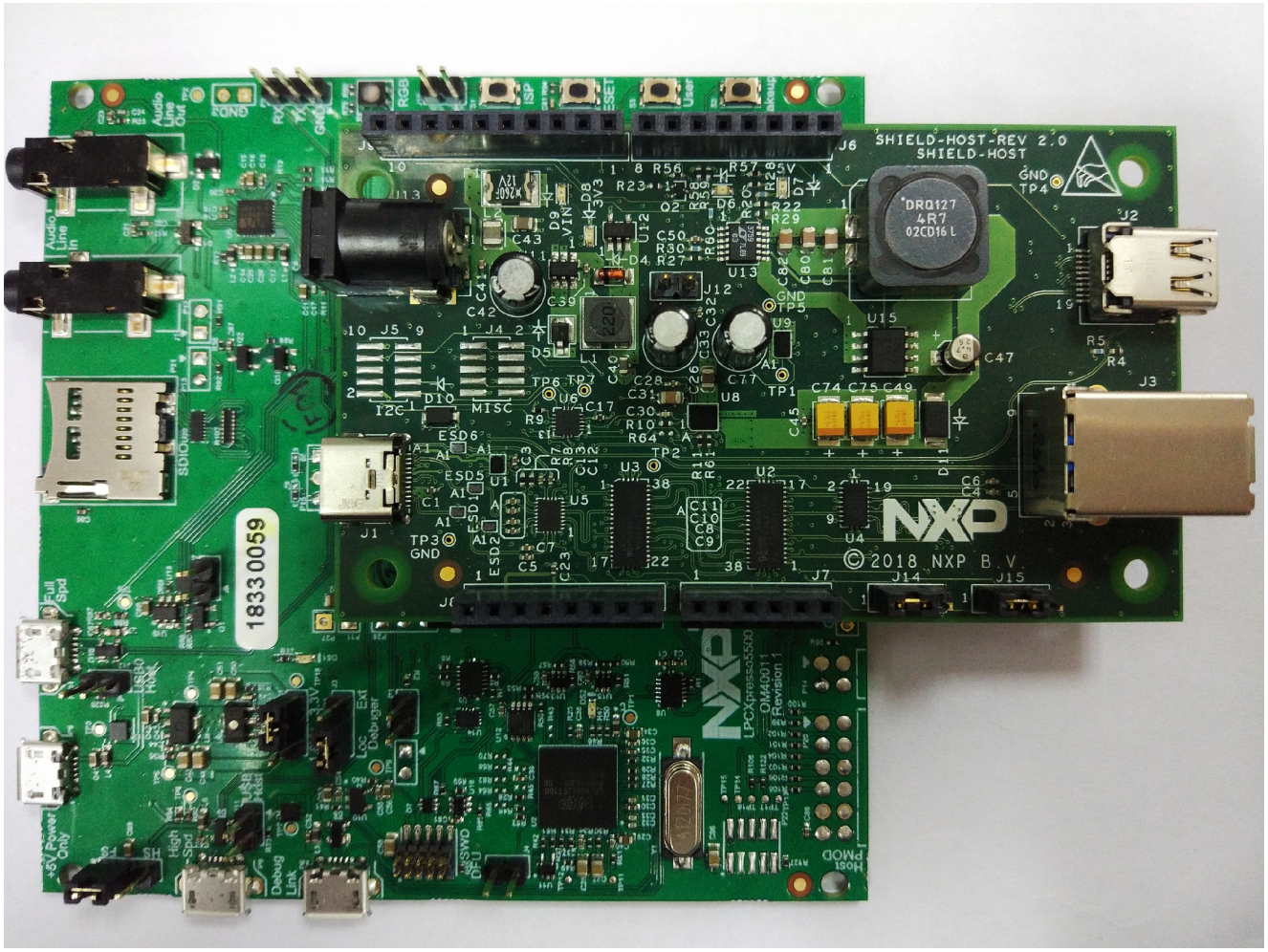

Figure 8. LPCXpresso55S69 connect shield board

FRDM-KL27Z: None

The connection state is as follows:

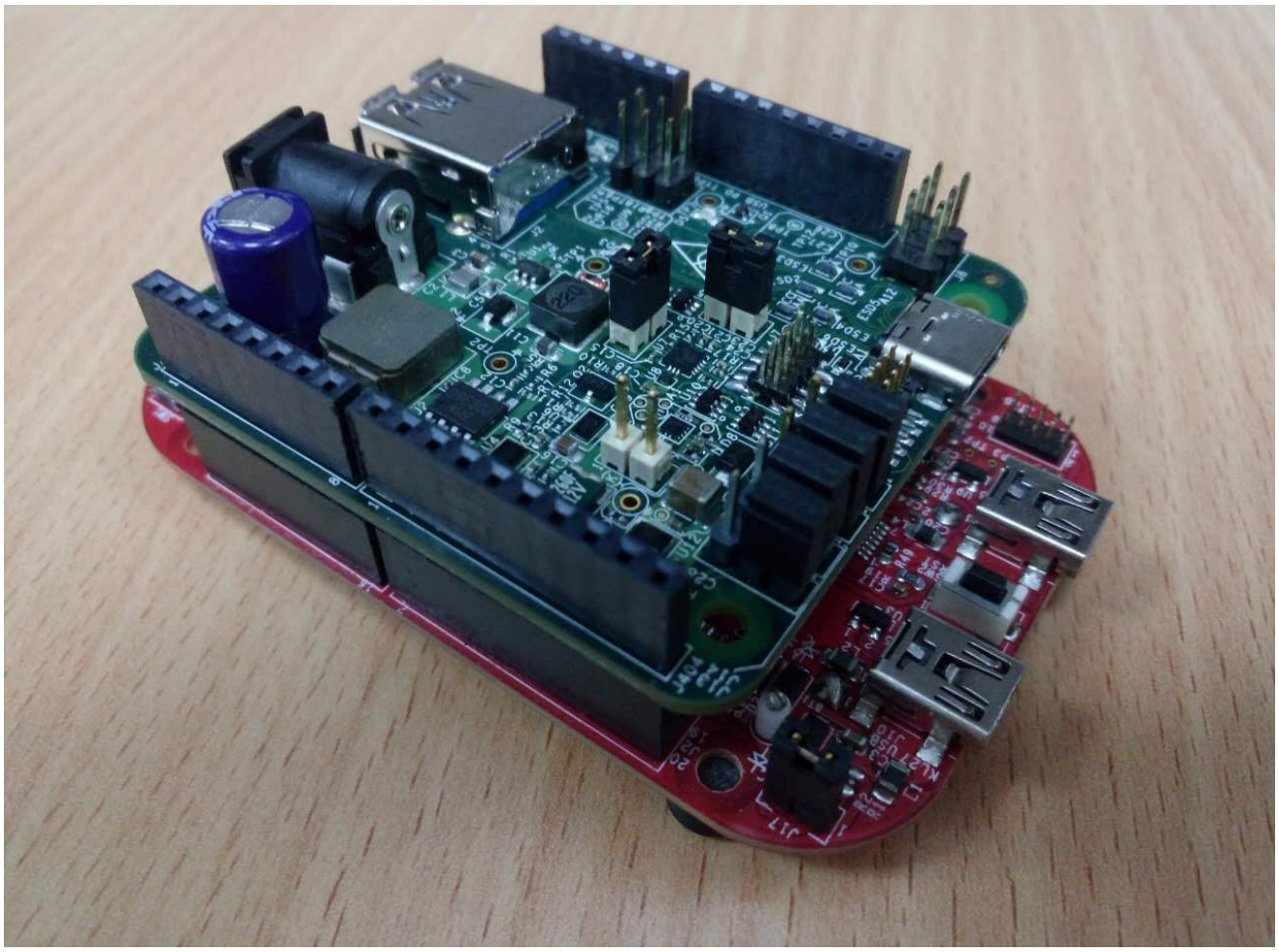

Figure 9. FRDM-KL27Z connect shield board

FRDM-K22F: None

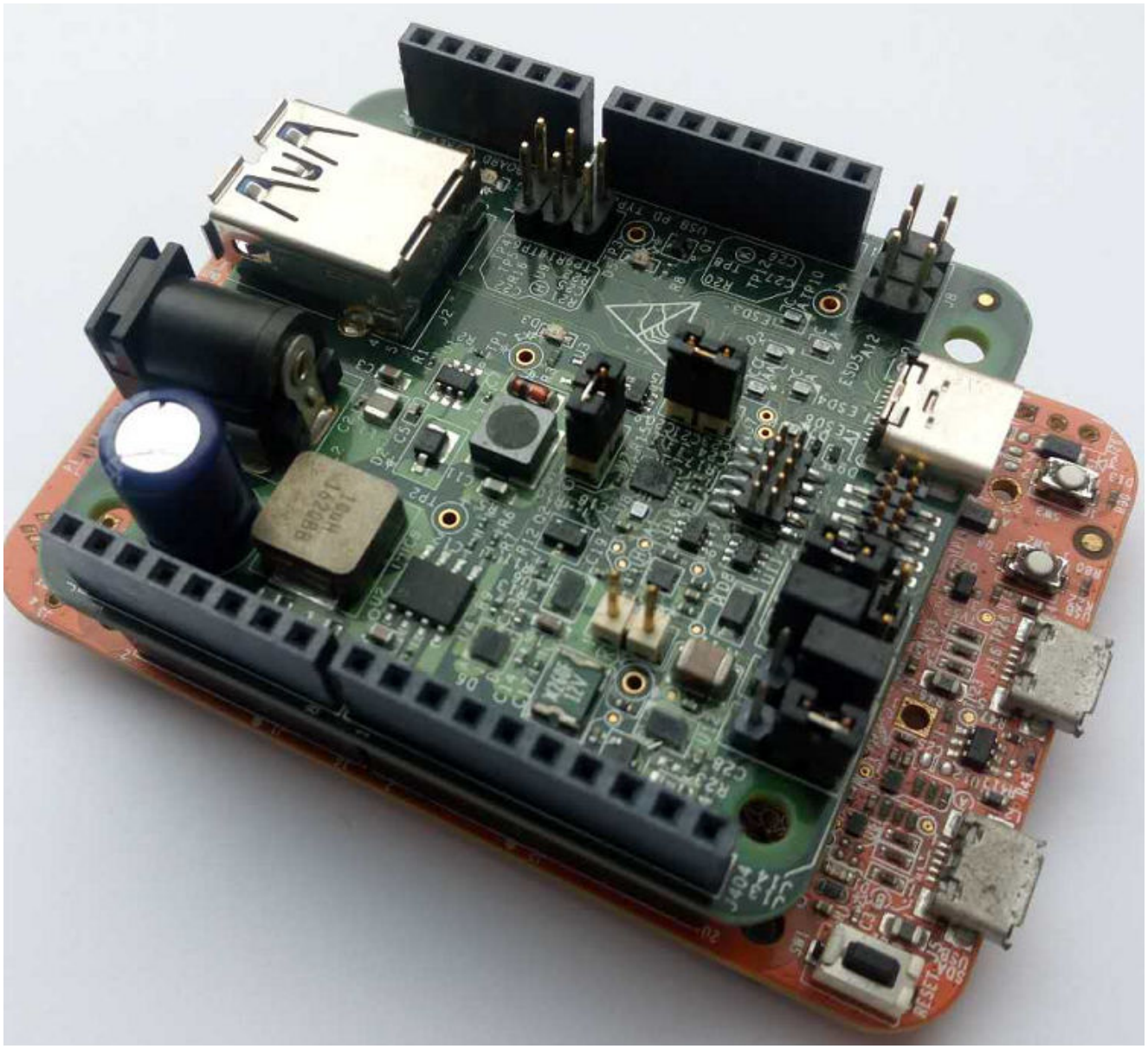

Figure 10. FRDM-K22F connect shield board

FRDM-K64F: None

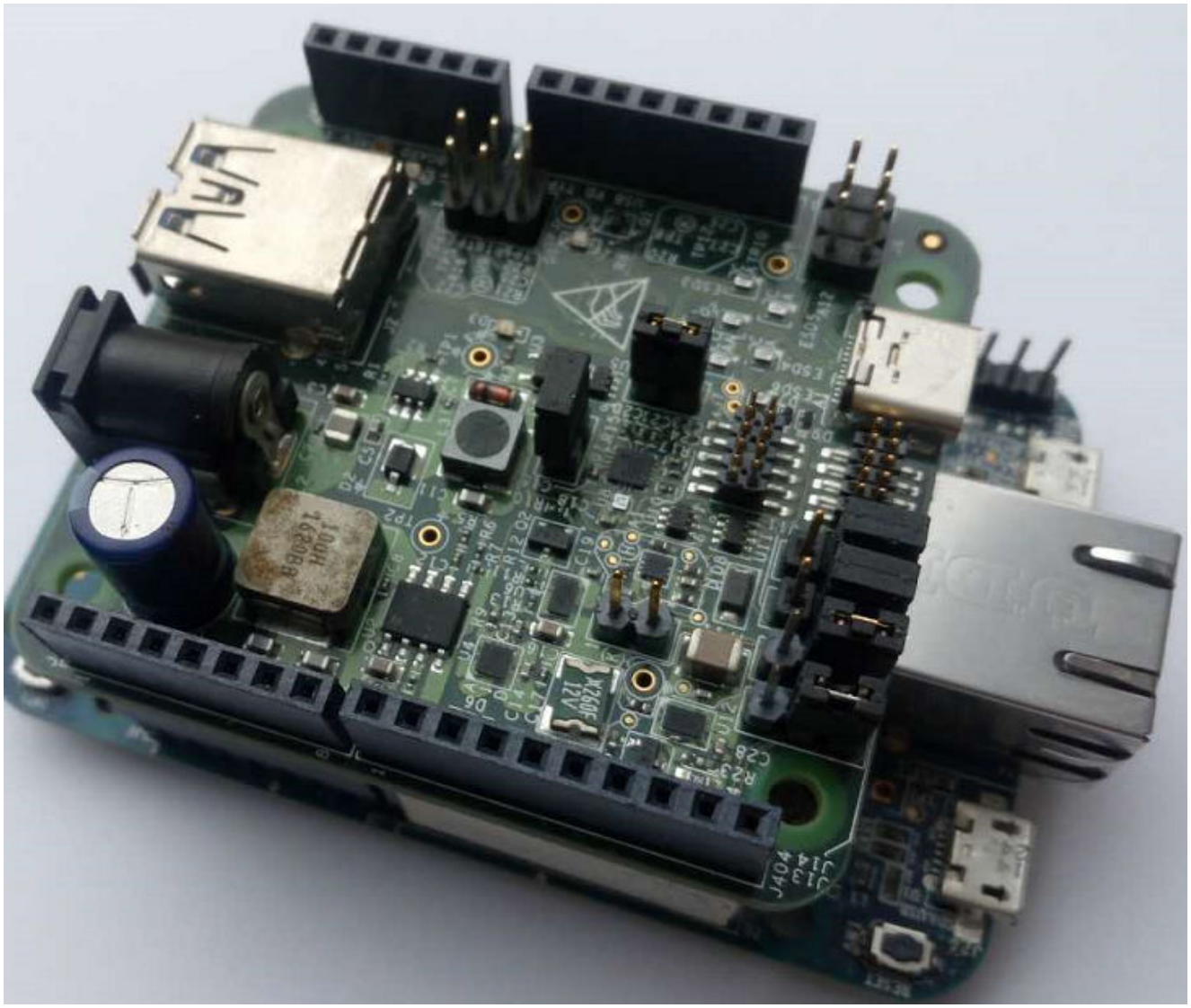

Figure 11. FRDM-K64F connect shield board

FRDM-KL28Z: None

The connection state is as follows:

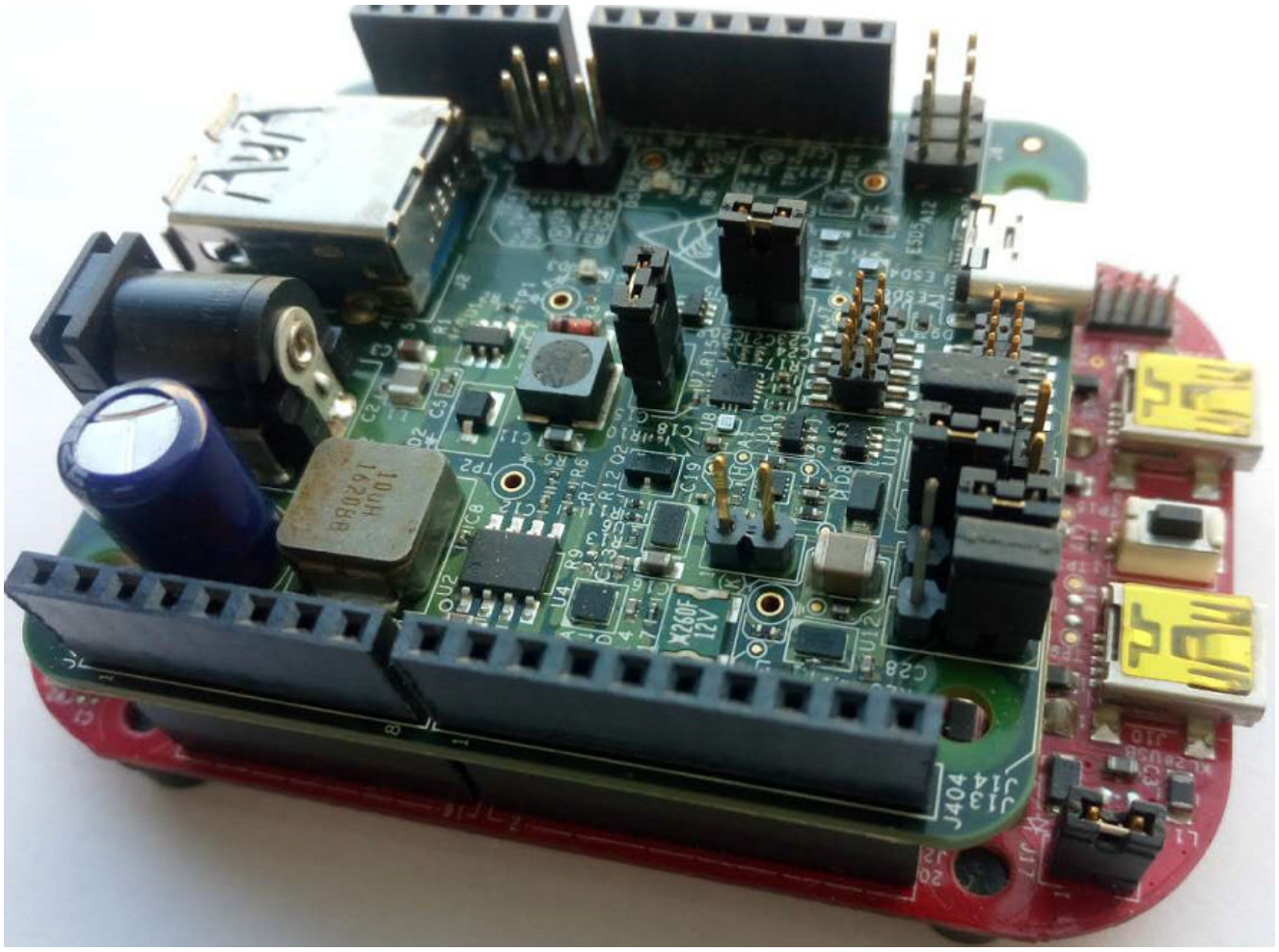

Figure 12. FRDM-KL28Z connect shield board

MIMXRT1050-EVK and IMXRT1050-EVKB: None

The connection state is as follows:

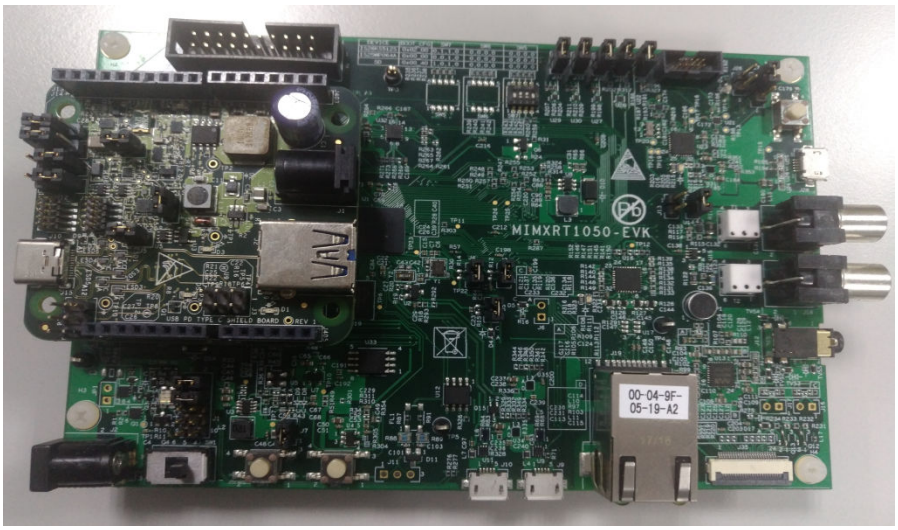

Figure 13. MIMXRT1050-EVK connect shield board

MIMXRT1020-EVK: None

The connection state is as follows:

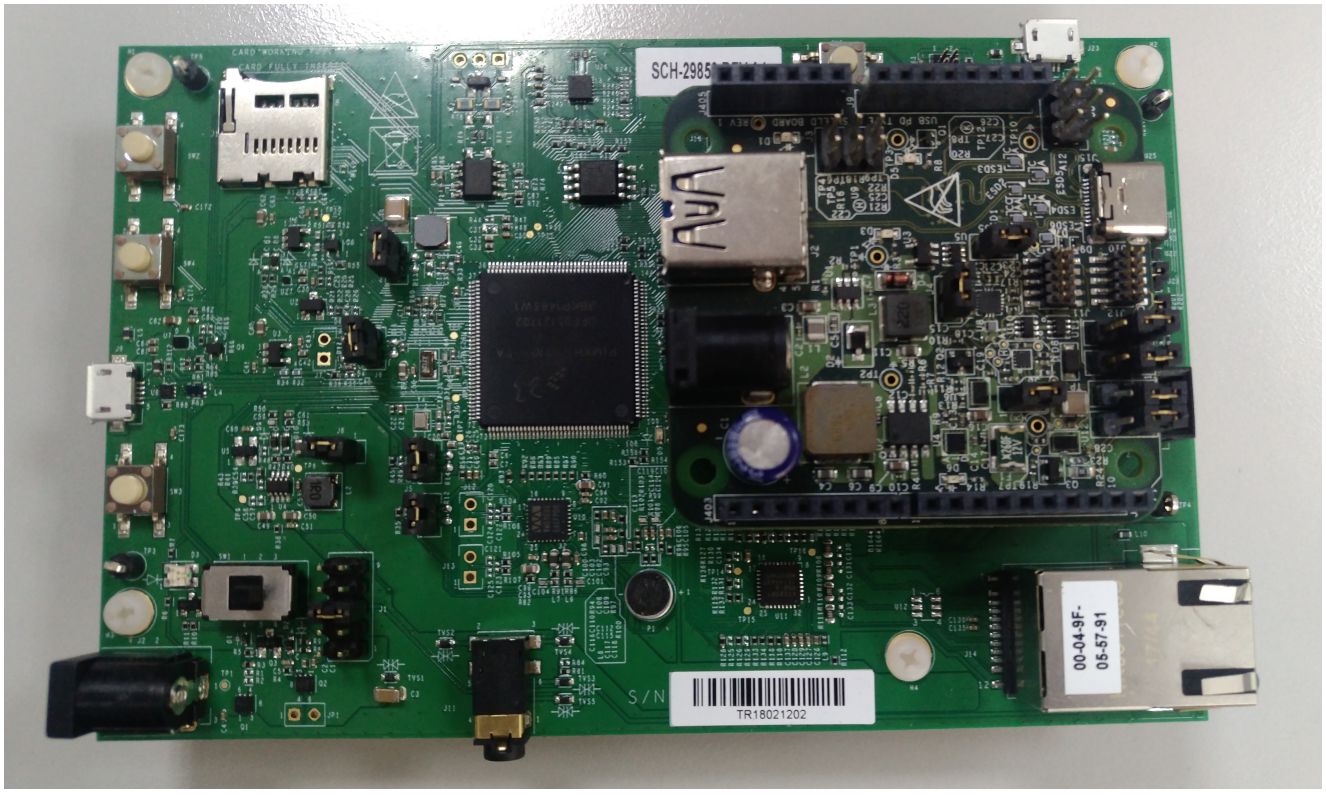

Figure 14. MIMXRT1020-EVK connect shield board

MIMXRT1060-EVK: None

The connection state is as follows:

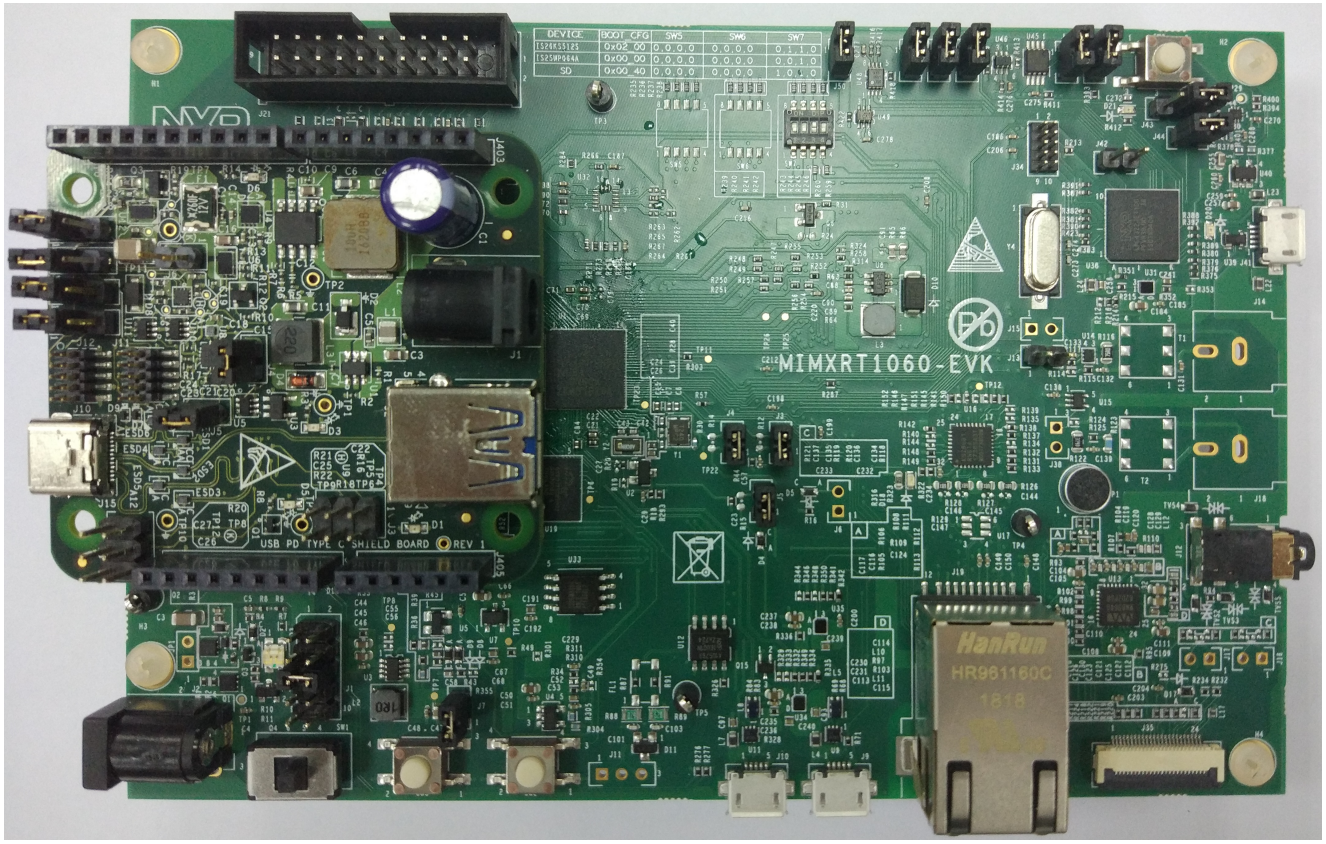

Figure 15. MIMXRT1060-EVK connect shield board

MIMXRT1064-EVK: None

The connection state is as follows:

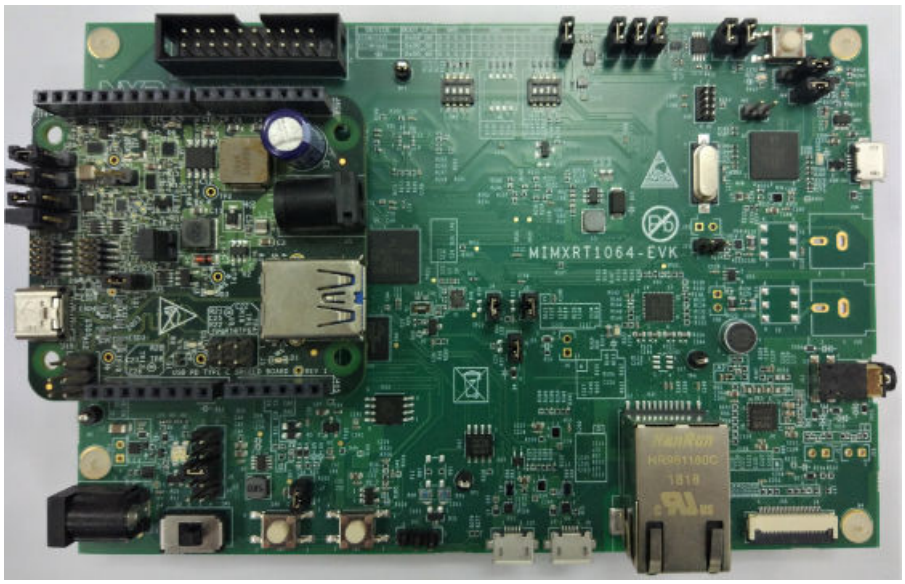

Figure 16. MIMXRT1064-EVK connect shield board

NOTE

Before powering up, verify the USB Type-C shield is pushed all the way into the Arduino header with good contact. Make sure there are no unintentional shorts (like jumper, blue wires) between the USB Type-C Shield board and the main development board.

MIMXRT685-EVK: None

The connection state is as follows:

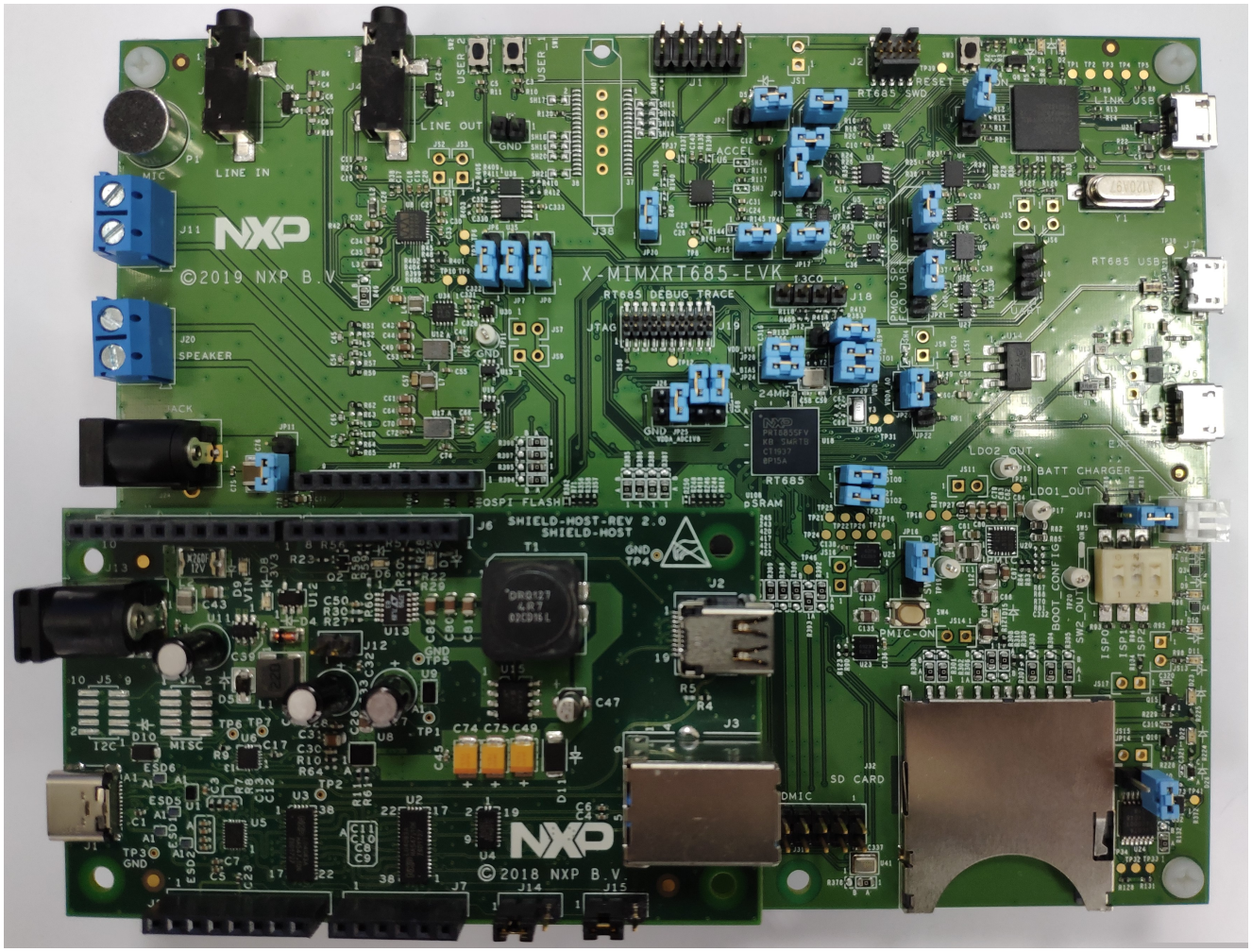

Figure 17. MIMXRT685-EVK connect shield board

MIMXRT1015-EVK: None

The connection state is as follows:

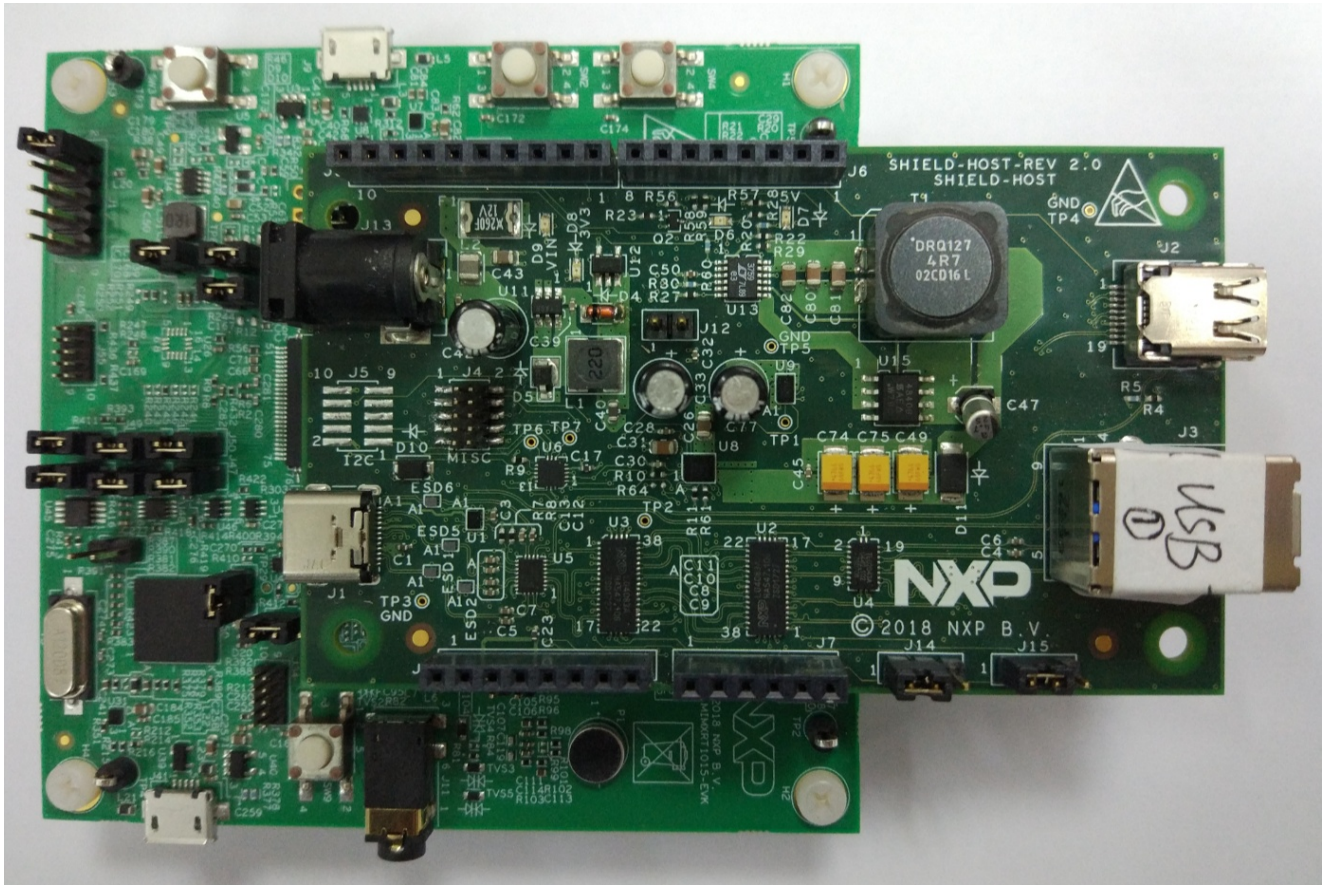

Figure 18. MIMXRT1015-EVK connect shield board

MIMXRT1170-EVK: None

The connection state is as follows:

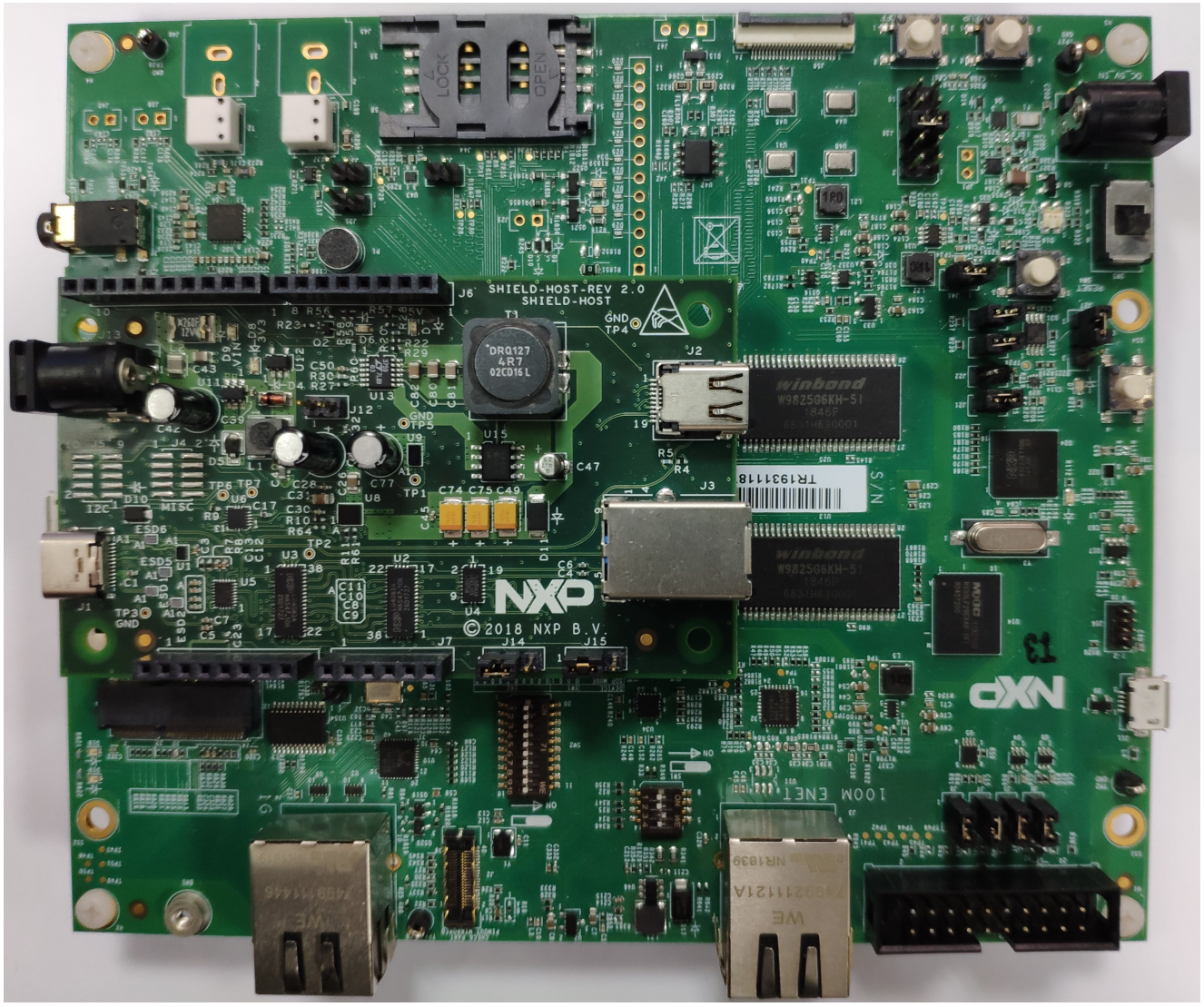

Figure 19. MIMXRT1170-EVK connect shield board

FRDM-KL32L2A4S: None

The connection state is as follows:

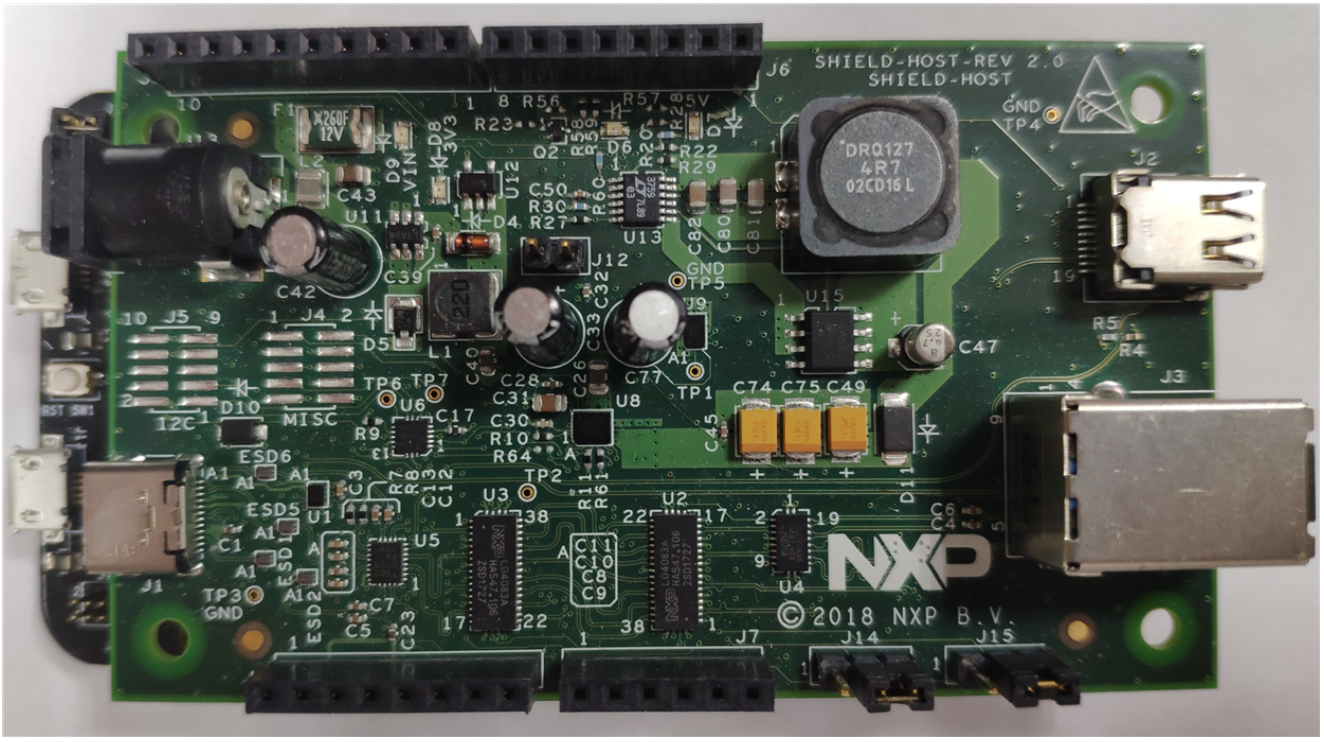

Figure 20. FRDM-KL32L2A4S connect shield board

LPCXpresso55S16: None

The connection state is as follows:

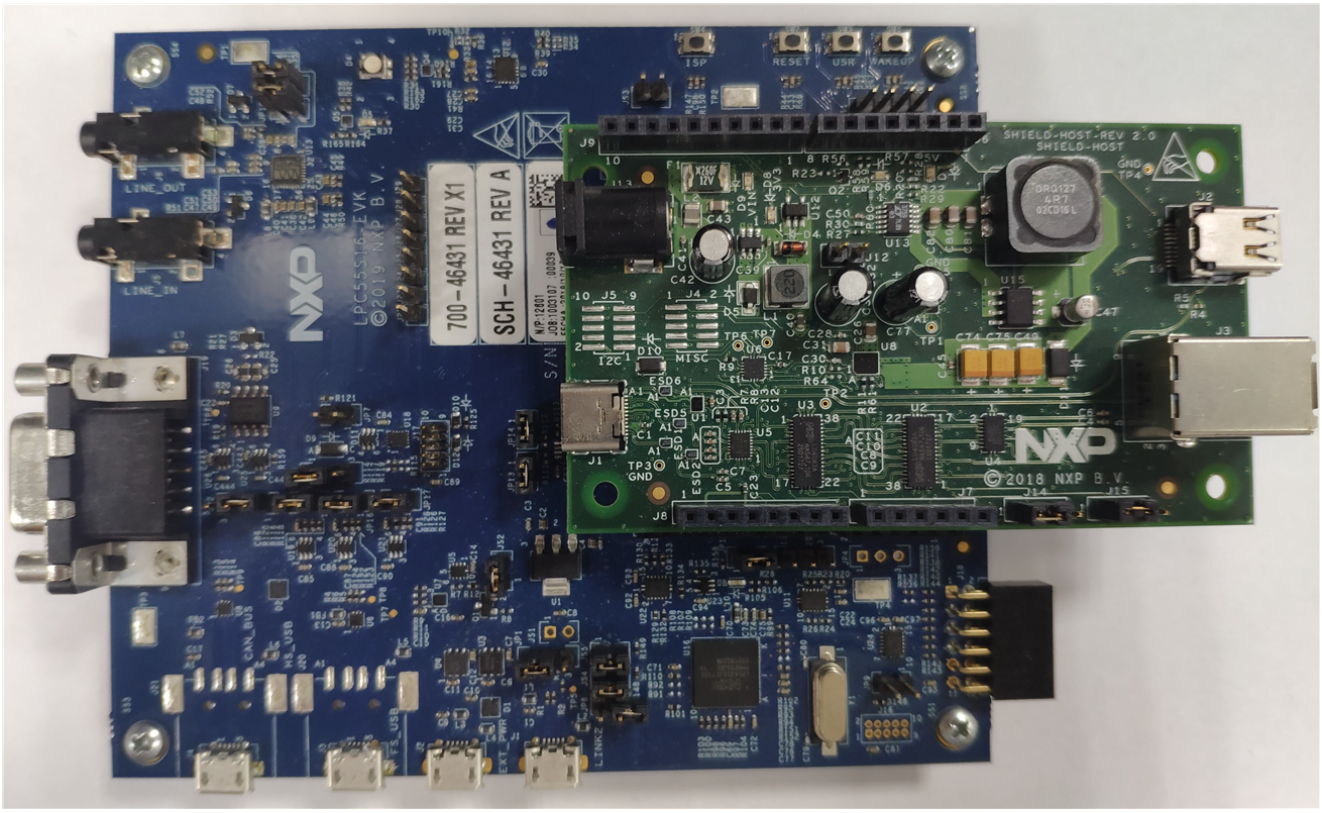

Figure 21. LPCpress55S16 connect shield board

MC56F83000-EVK: None

The connection state is as follows:

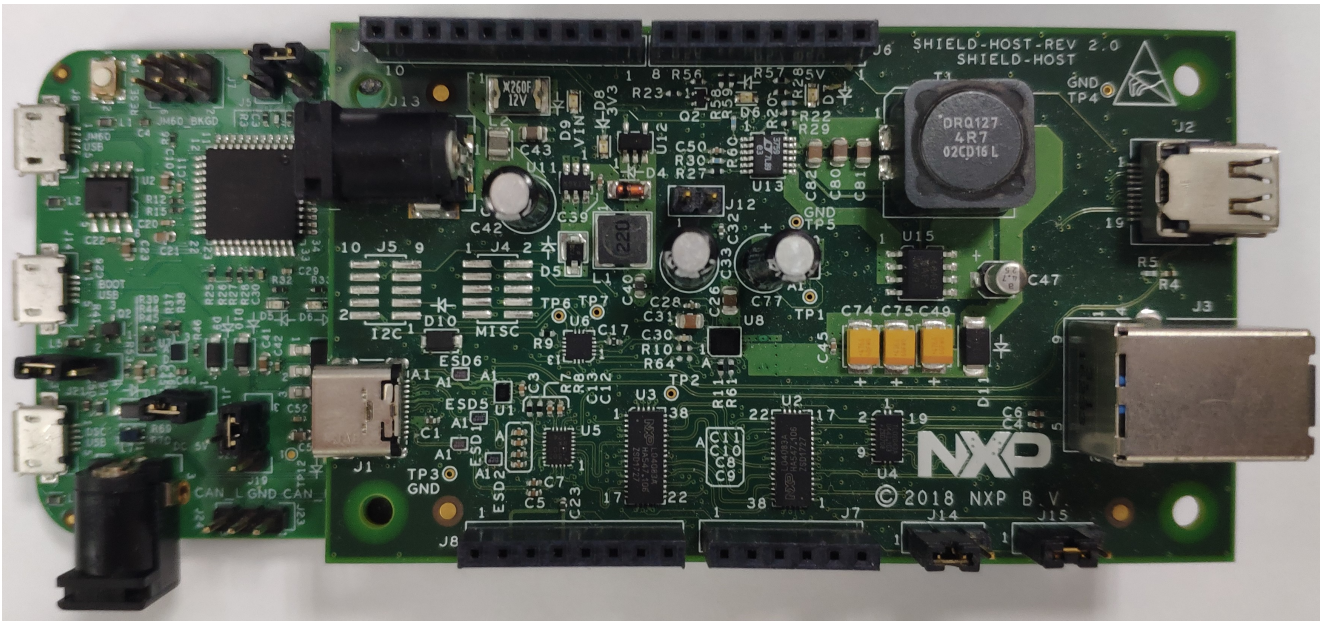

Figure 22. MC56F83000-EVK connect shield board

MIMXRT1060-EVKB: None

The connection state is as follows:

Figure 23. MIMXRT1060-EVKB connect shield board

MIMXRT1160-EVK: None

The connection state is as follows:

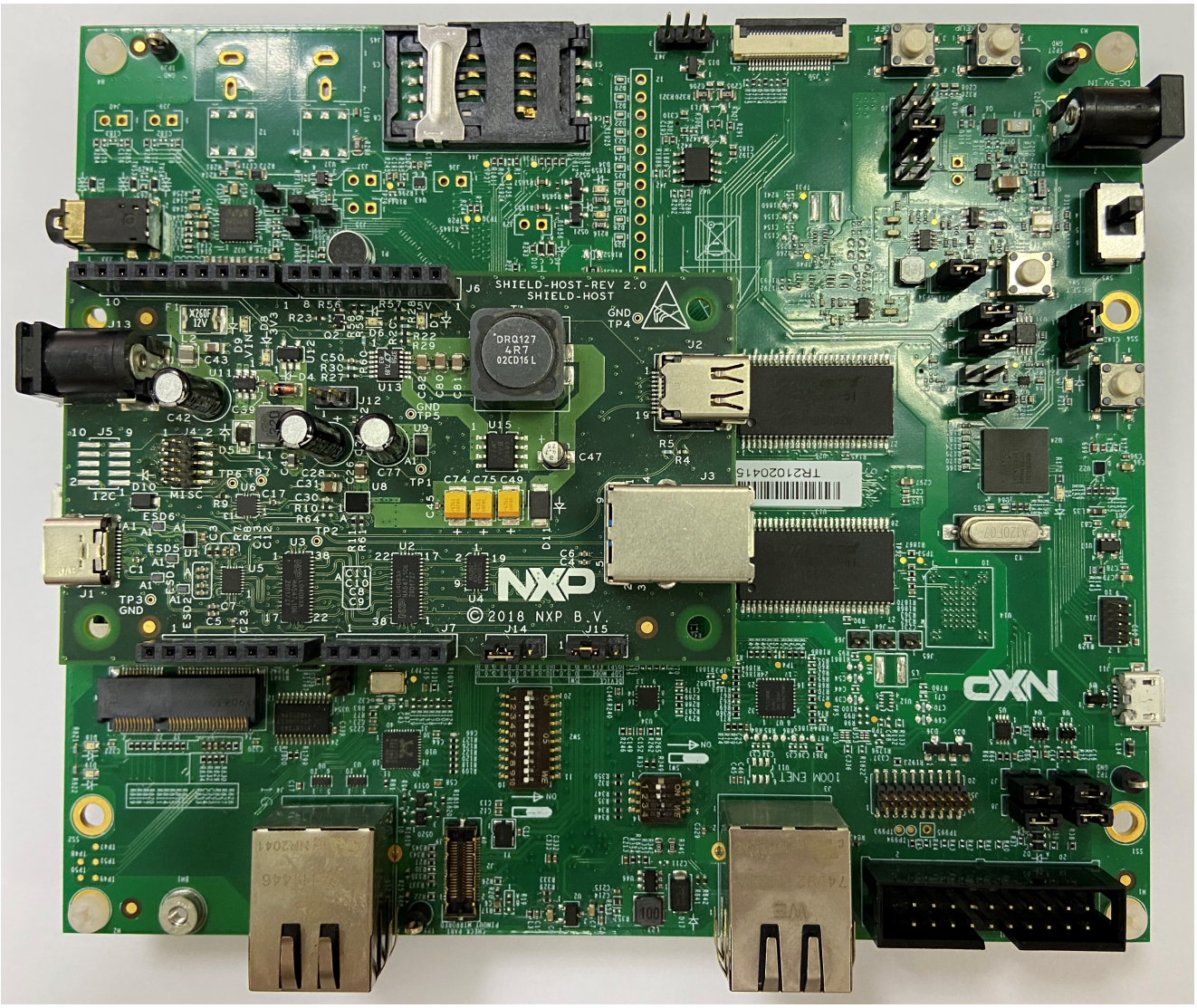

Figure 24. MIMXRT1160-EVK connect shield board

MIMXRT595-EVK: None

The connection state is as follows:

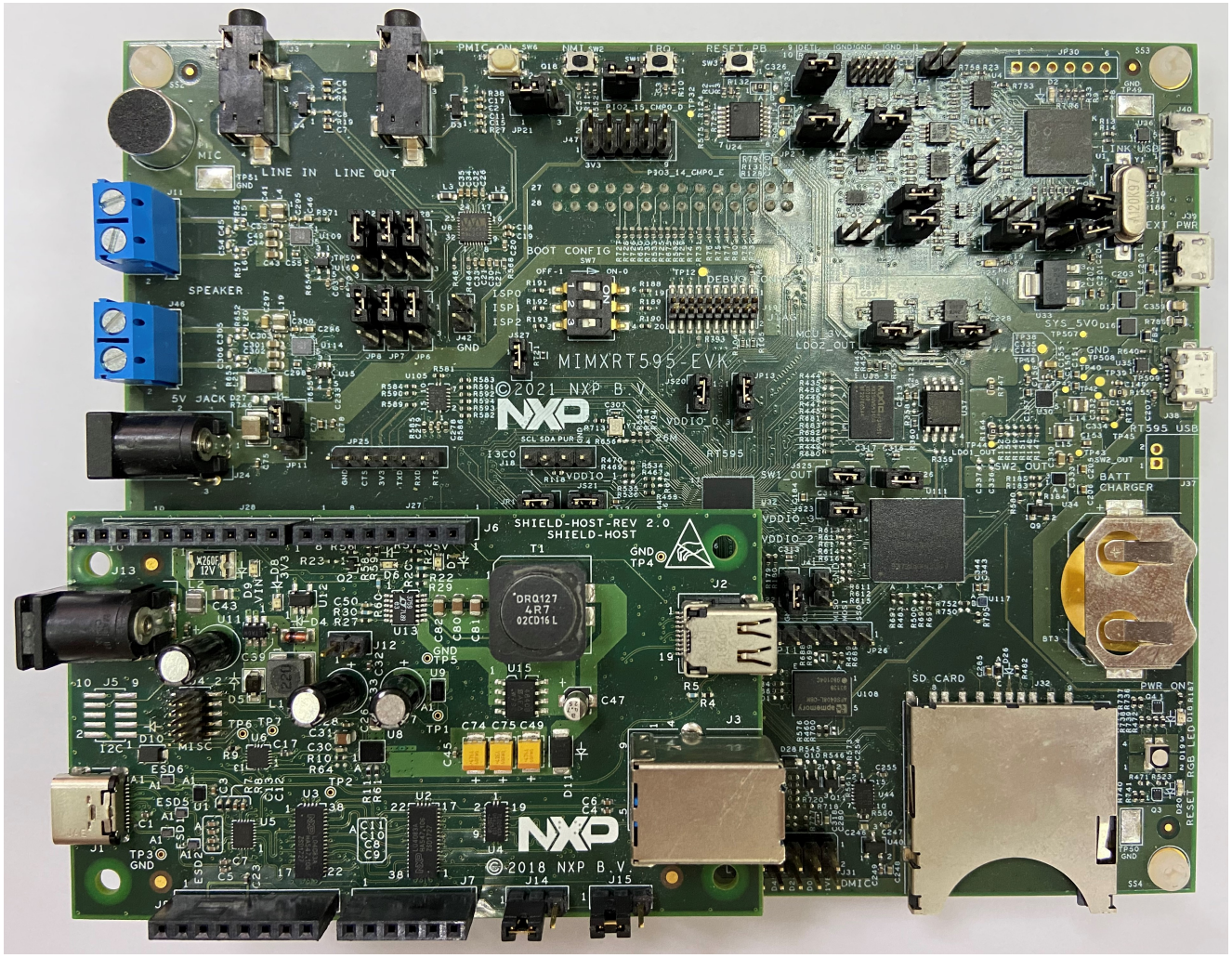

Figure 25. MIMXRT595-EVK connect shield board

Chapter 4
Demo setup

The `usb_pd` demo is described in this section. For information about `usb_pd_battery`, `usb_pd_source_charger`, and `usb_pd_alt_mode_dp_host`, see the *readme* file in the corresponding demos' directory.

The LPCXpresso54114 and USB-PD/Type C shield board om13588 is used as the example.

In this demo, several functionalities are demonstrated: Power Role Swap, Request 5 V or higher voltage, Hard Reset, and so on. There are two ways to control the demo, using the button or user menu. Due to the limitation of number of hardware buttons, additional functionalities are provided via user menu. Table 3 indicates the way it is used on different boards:

Table 3. Software for board

	Power request software	Power change software
FRDM-K22F	SW2	SW3
FRDM-K64F	SW2	SW3
FRDM-KL27Z	SW1	SW3
FRDM-KL28Z	user menu	user menu
FRDM-KL32L2A4S	user menu	user menu
IMXRT1050-EVKB	user menu	user menu
LPCXpresso54018	SW4	SW5
LPCXpresso54114	SW1	SW2
LPCXpresso54608	SW4	SW5
LPCXpresso55S16	user menu	user menu
LPCXpresso55S28	user menu	user menu
LPCXpresso55S69	user menu	user menu
MIMXRT685-EVK	user menu	user menu
MIMXRT1015-EVK	user menu	user menu
MIMXRT1020-EVK	user menu	user menu
MIMXRT1050-EVK	user menu	user menu
MIMXRT1060-EVK	user menu	user menu
MIMXRT1064-EVK	user menu	user menu
MC56F83000-EVK	user menu	user menu
MIMXRT1170-EVK	user menu	user menu
MIMXRT1060-EVKB	user menu	user menu
MIMXRT1160-EVK	user menu	user menu
MIMXRT595-EVK	user menu	user menu

NOTE

For some boards there are no switches to use. So, this demo uses menus to implement the same functionality as a switch. The menus correspond to the switches as follows and can be obtained by inputting **0** in the debug console.

Table 4. Menus for switch functions

Switch	Menu item
Short press **Power request switch**	Request 5 V
Long press **Power request switch**	Request high voltage
Short press **Power change switch**	Power role swap
Long press **Power change switch**	Hard reset

NOTE

The VBus test point is **J5** of the USB-PD/Type C Shield board.

4.1 Setup hardware boards

1. Connect the debug console port to PC. For example, connect J7 of LPCXpresso54114 to the PC.
2. Set the shield board's jumpers and connect the shield board with the development board as shown in section Hardware rework.
3. Connect a 9 V power source to the J1 jack in the USB-PD/Type C Shield board. The shield LEDs status is as follows:

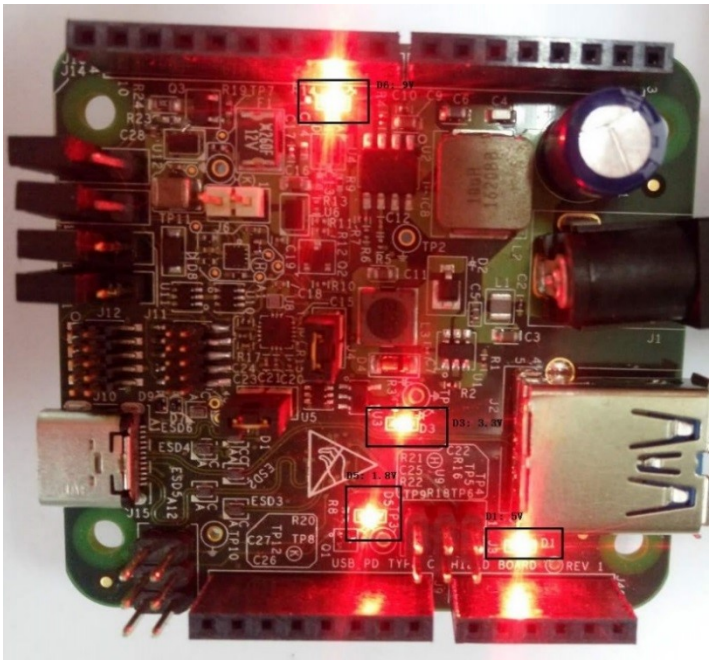

Figure 26. Shield board LEDs

4. Ensure you run the usb_pd demo by using the instructions in section Running the demo.
5. "pd init success" prints in the debug console.

4.2 Request from original sink role

1. Connect a USB Type-C cable between two boards. One works as a sink and one works as a source. You can see it through the debug console.
2. Connect a voltmeter to VBus (J5) of the sink role board. The voltmeter at the sink role shows approximately 5 V.

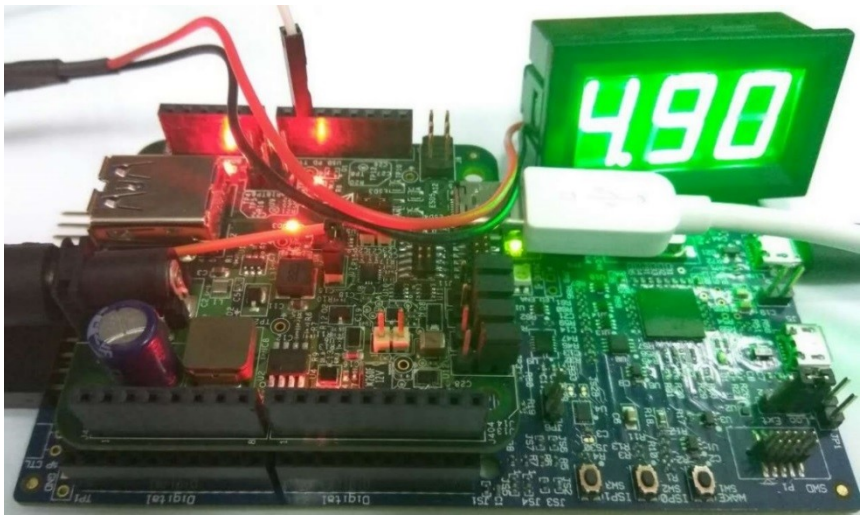

Figure 27. 5 V VBus

3. Press “Power request switch ” for about 3 seconds to make 9 V request. After the request is completed successfully, the voltmeter shall show 9 V.

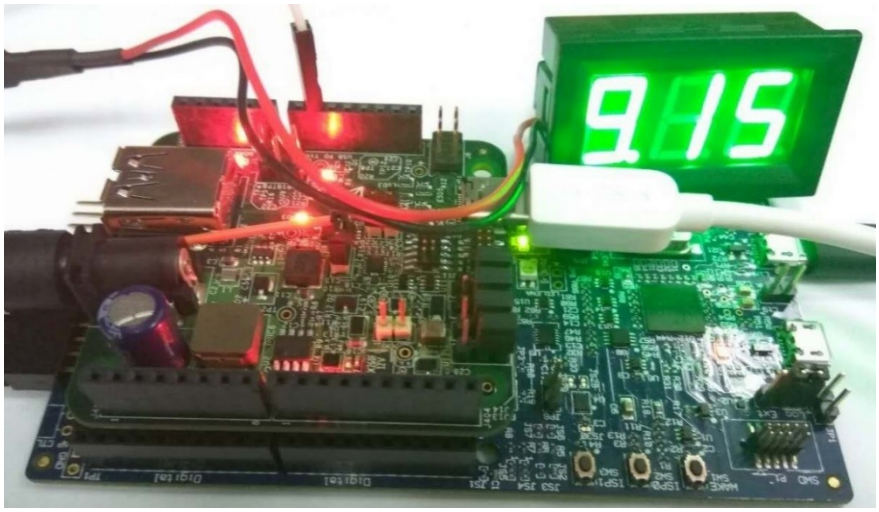

Figure 28. 9 V VBus

4. Immediately press “Power request switch” to make 5 V request. After the request is completed successfully, the voltmeter shows approximately 5 V.

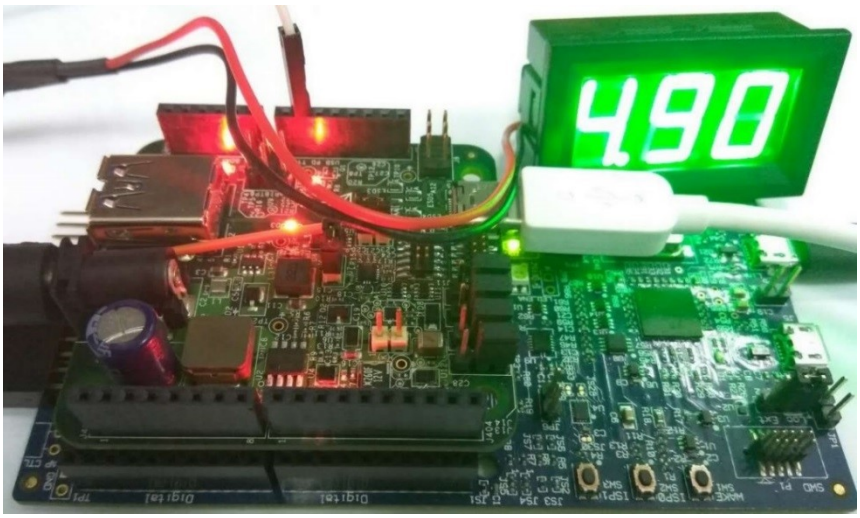

Figure 29. 5 V VBus

4.3 Power swap from sink role

1. Immediately press the “Power change switch” on the sink to make PR_SWAP.
2. The voltage of the VBus drops to 0 V, then back to 5 V.

4.4 Request from original source role

1. Connect a voltmeter to VBus (J5) of the new sink role board. After the power role swap, the original source role becomes the sink role. The voltmeter at the source role shows approximately 5 V.

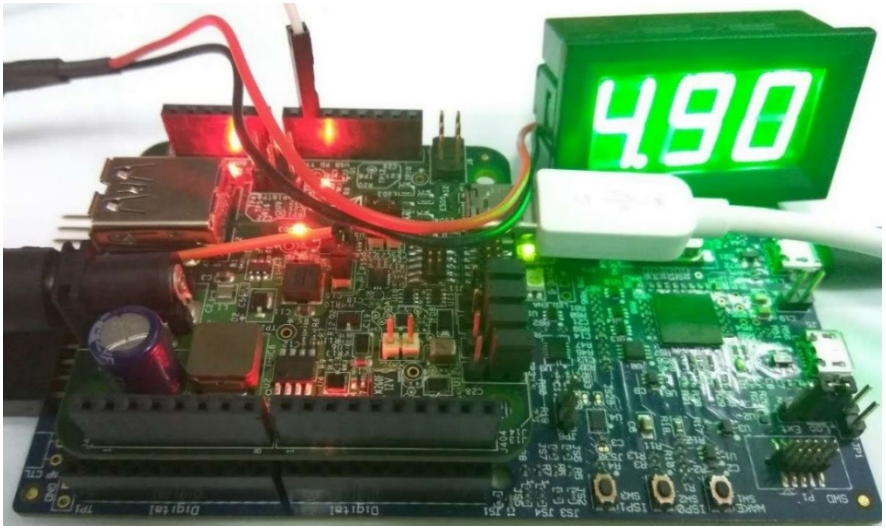

Figure 30. 5 V VBus

2. Press the “Power request switch ” for about 3 seconds to make 9 V request. After the request is completed successfully, the voltmeter shows approximately 9 V.

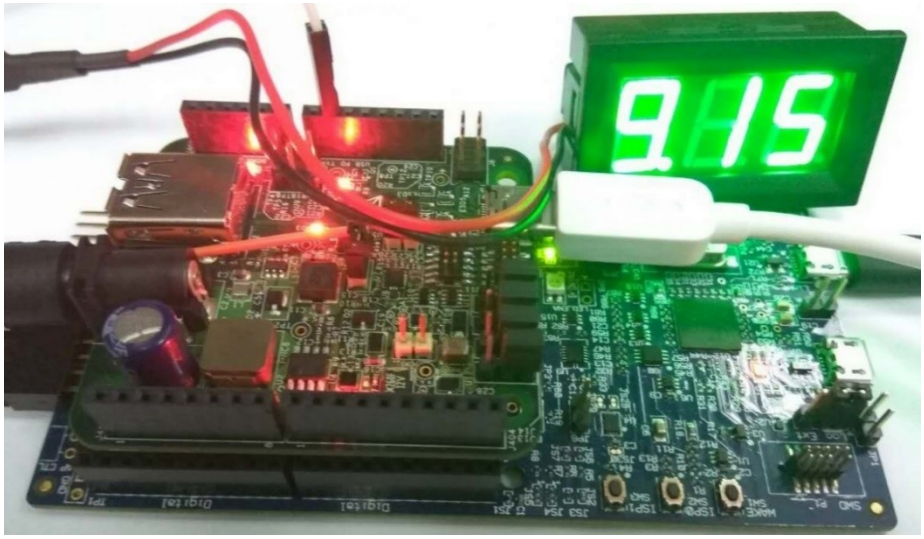

Figure 31. 9 V VBus

3. Immediately press the “Power request switch ” to make 5 V request. After the request is completed successfully, the voltmeter shows approximately 5 V.

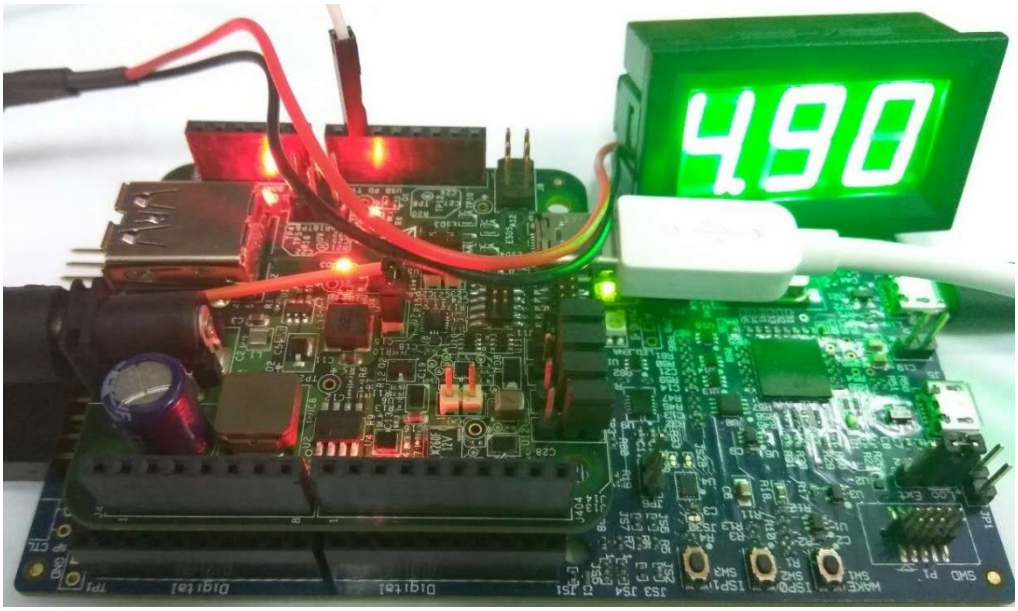

Figure 32. 5 V VBus

4.5 Power swap from source role

1. Immediately press “power change switch ” on the source to make PR_SWAP.
2. The voltage of the VBus drops to 0 V, then back to 5 V.

4.6 Hard reset test

1. Press the “Power change switch ” for 3 seconds to make HARD_RESET.
2. The Voltage of the VBus drops to 0 V, then back to 5 V.

3. The source and sink state machine restart, and sink requests the power again. This can be seen in the logs in the debug console.

4.7 Test other commands

1. Input ‘0’ in the debug console. The following menu is printed in the debug console (the menu is a little different for source and sink):

```
The menu is as follow for source:
0. print menu
a. source power change
b. goto min
c. fast role swap
e. soft reset
f. data role swap
g. vconn swap
h. get partner sink capabilities
i. standard structured VDM test (only DFP can send enter mode)
j. exit mode (only DFP)
k. send attention
l. test vendor structured VDM
m. test unstructured VDM
n. get source extended capabilities
o. get status
p. alert
q. get battery capabilities
r. get battery status
s. get manufacturer info
t. cable reset  (not supported yet)
```

Figure 33. Commands menu

2. Input the menu to test the corresponding command.
3. For example, if you input ‘f’, the “data role swap” command begins, and the debug console prints the result.

Chapter 5
PD compliance test

Ellisys test environment

- Hardware: Ellisys EX350
- Software: Ellisys USB Explorer 350 Examiner 3.1.7812

The usb_pd example supports the PD3.0 compliance test. The usb_pd_alt_mode_dp_host example supports DisplayPort host alternate mode compliance test. The usb_pd_alt_mode_dp_dock example supports DisplayPort dock alternate mode compliance test.

The tested target for PD3.0 compliance test is usb_pd_freertos flexspi_nor_release target of IAR toolchain on the MIMXRT1064-EVK board.

The tested target for DisplayPort host alternate mode compliance test is usb_pd_alt_mode_dp_host_freertos flexspi_nor_release target of IAR toolchain on the MIMXRT1064-EVK board.

The tested target for DisplayPort dock alternate mode compliance test is usb_pd_alt_mode_dp_dock_freertos release target of IAR toolchain on the LPCXpresso55S69 board.

The test reports are in *boards/middleware/usb/pd/compliance_test_report*.

To do a compliance test, the following configurations need be enabled in the usb_pd_config.h file.

PD_CONFIG_COMPLIANCE_TEST_ENABLE, PD_CONFIG_TRY_SNK_SUPPORT and PD_CONFIG_TRY_SRC_SUPPORT.

For the usb_pd example, five configurations are verified and passed: correspondingly, one dedicated MACRO and one VIF file are needed to enable each of them. The MACROs are defined in the pd_board_config.h and the VIF files are in *boards/<board>/usb_examples/usb_pd/<bm or freertos>/VIF*:

- Dual-role port

  PD_COMPLIANCE_TEST_DRP needs to be enabled in pd_board_config.h and the file drp.txt is the dedicated VIF file for this test.

- Dual-role port with Try.SNK

  PD_COMPLIANCE_TEST_DRP_TRY_SNK needs to be enabled in pd_board_config.h and the file drp_try_snk.txt is the dedicated VIF file for this test.

- Dual-role port with Try.SRC

  PD_COMPLIANCE_TEST_DRP_TRY_SRC needs to be enabled in pd_board_config.h and the file drp_try_src.txt is the dedicated VIF file for this test.

- Consumer/Provider port

  PD_COMPLIANCE_TEST_CONSUMER_PROVIDER needs to be enabled in pd_board_config.h and the file consumer_provider.txt is the dedicated VIF file for this test.

- Provider/Consumer port

  PD_COMPLIANCE_TEST_PROVIDER_CONSUMER needs to be enabled in pd_board_config.h and the file provider_consumer.txt is the dedicated VIF file for this test.

NOTE

Only one macro can be enabled at a time for these five configurations.

For the usb_pd_alt_mode_dp_host and usb_pd_alt_mode_dp_dock example, the MACROs are defined in the pd_board_config.h and do not need to be modified. The corresponding VIF file is in boards/<board>/usb_examples/ usb_pd_alt_mode_dp_host /<bm or freertos>/VIF and boards/<board>/usb_examples/ usb_pd_alt_mode_dp_dock /<bm or freertos>/VIF

Chapter 6
Known issues

Five issues were found during the Ellisys compliance test as shown below.

- Hardware: Ellisys EX350
- Software: Ellisys USB Explorer 350 Examiner 3.1.7812

Issue1: TD 4.3.1 Sink Connect Source

Issue2: TD 4.3.2 Sink Connect DRP

Issue3: TD 4.3.3 Sink Connect Try.SRC DRP

Issue4: TD 4.3.4 Sink Connect Try.SNK DRP

For issues 1-4. these items will fail when using the consumer_provider and displayport_dock VIF file. The om13790host and om13790dock have these failures and om13588 does not have these failures. The reason for these failures is that Ellisys EX350 will pull up a 5 V pulse with a fast rising edge on CC lines before starting testing these items, which triggers NX20P0407 OVP protection. NX20P0407 is USB Type C CC and SBU Protection IC. To pass these test items, solder a 100pF capacitor between CC lines and ground at the connector to slow down the fast rising edge as a workaround. We are working with Ellisys to eliminate this 5 V pulse.

Chapter 7
Revision history

This table summarizes revisions to this document.

Table 5. Revision history

Revision number	Date	Substantive changes
1.0	04/2017	USB PD release for FRDM-KL27Z (based on SDK2.2 package)
1.1	05/2017	USB PD for LPCXpresso54608 (based on SDK2.2 package)
1.2	05/2017	USB PD release for LPCXpresso54114
1.3	05/2017	USB PD release for LPCXpresso54608
1.4	06/2017	USB PD release for KL27Z
1.5	10/2017	Added FRDM-K22F, FRDM-K64F, and FRDM-KL28Z support
1.6	10/2017	Added MIMXRT1050-EVK support
1.7	05/2018	MCUXpresso SDK v2.4.0 release
1.8	08/2018	Added MIMXRT1060-EVK support
1.9	08/2018	Added MIMXRT1064-EVK support
2.0	11/2018	MCUXpresso SDK v2.5.0 release
2.1	12/2018	Added LPCXpresso55S69 support
2.2	01/2019	Added MIMXRT685-EVK support
2.3	03/2019	• Added MIMXRT1015-EVK support • Updated MIMXRT1050-EVK, IMXRT1050-EVKB, MIMXRT1020-EVK, MIMXRT1060-EVK, MIMXRT1064-EVK, MIMXRT685-EVK, and MIMXRT1015-EVK to correct board names throughout document
2.4	06/2019	MCUXpresso SDK v2.6.0 release
2.5	08/2019	Added LPCXpresso55S28 support
2.6	12/2019	MCUXpresso SDK v2.7.0 release
3	05/2020	MCUXpresso SDK v2.8.0 release
4	09/2020	Add description for MC56F83000-EVK
5	11/2020	MCUXpresso SDK v2.9.0 release
6	10 July 2021	Updated for MCUXpresso SDK v2.10.0 release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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.

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 10 July 2021

Document identifier: MCUXSDKUSBPDUG

arm