Document identifier: IMX8MPPBUG
Rev. 0, 29 April 2021

i.MX 8M Plus Power Measurement Board User's Guide

Contents

Chapter 1
Introduction

This is the user’s guide for the 8MPLUSLPD4-PEVK board based on the i.MX 8M Plus applications processor. It includes the hardware/software setup and configurations and detailed information on the overall design.

Chapter 2
Board overview

The 8MPLUSLPD4-PEVK consists of two separate boards: 8MPLUSLPD4-PWR (CPU board that supports power measurement) and 8MPLUS-BB (Base Board). It is software-compatible with the 8MPLUSLPD4-EVK. The 8MPLUSLPD4-PEVK board can measure 27 power rails, including 7 power rails that have dual-range options, for low-power measurement. All PHY power rails are separated for measurement, so you can conveniently obtain the power consumption for each analog PHY.

2.1 Power rails

Table 1 lists the detailed information of the tested rails.

Table 1. Power rails under measurement

NO.		Seq.	Power rail	Operating range (V)			Monitor number	Range1 (mA)	Rs1 (OHM)	Range2 (mA)	Rs2 (OHM)
				Min	Normal	Max					
1	C	1	VSYS_5V	-			M2	5000	0.02	238	0.42
2	J	2	NVCC_SNVS_1V8	1.65	1.8	1.950	M4	10	10	0.2	509
3	D	3	VDD_SOC	0.9	0.95	1	M2	10000	0.01	100	1.01
4	T		VDD_PLL_ANA_0V8	0.9	0.95	1	M7	100	1	-	-
5	W		VDD_HDMI_0V8	0.9	0.95	1	M7	100	1	-	-
6	V		VDD_MIPI_0V8	0.9	0.95	1	M7	100	1	-	-
7	U		VDD_PCI_0V8	0.805	0.85	0.9	M7	100	0.4	-	-
8	Y		VDD_USB_0V8	0.9	0.95	1	M8	250	0.4	-	-
9	A	4	VDD_ARM	0.95	1	1.05	M1	5000	0.02	20	5.01
10	O	5	VDD_PLL_ANA_1V8	1.71	1.8	1.89	M5	400	1.008	9.1	11
11	K		VDD_EARC_1V8	1.71	1.8	1.89	M4	50	2	-	-
12	I		VDD_HDMI_1V8	1.71	1.8	1.89	M4	50	2	-	-
13	H		VDD_LVDS_1V8	1.71	1.8	1.89	M4	100	1	-	-
14	N		VDD_MIPI_1V8	1.71	1.8	1.89	M5	50	2	-	-
15	M		VDD_PCI_1V8	1.71	1.8	1.89	M5	100	1	-	-
16	L		VDD_USB_1V8	1.71	1.8	1.89	M5	50	2	-	-
17	R	6	CPU_VDD_1V8	1.65	1.8	1.95	M6	1000	0.1	-	-
18	P		NVCC_SD1	1.65	1.8	1.95	M6	1000	0.1	-	-

Table continues on the next page...

Table 1. Power rails under measurement (continued)

NO.		Seq.	Power rail	Operating range (V)			Monitor number	Range1 (mA)	Rs1 (OHM)	Range2 (mA)	Rs2 (OHM)
				Min	Normal	Max					
19	*Q*		LPD4_VDD1	1.7	1.8	1.95	M6	20	4.99	-	-
20	*S*		BB_VDD_1V8	1.65	1.8	1.89	M6	2000	0.05	-	-
21	*B*	*7*	*NVCC_DRAM_1V1*	*1.045*	*1.1*	*1.155*	M1	*2000*	*0.05*	*48.8*	*2.05*
22	*E*		*LPD4_VDDQ*	*1.06*	*1.1*	*1.17*	M3	*2000*	*0.05*	*48.8*	*2.05*
23	*F*		LPD4_VDD2	1.06	1.1	1.17	M3	*2000*	*0.05*	-	-
24	*X*	*8*	VDD_USB_3V3	3.069	3.3	3.63	M8	100	1	-	-
25	*Z*		VDD_SD1_3V3	2.7	3.3	3.6	M8	400	0.25	-	-
26	*a*		BB_VDD_3V3	3	3.3	3.6	M8	2000	0.05	-	-
-		*9*	BB_VSD_3V3	-			-	-	-	-	-
27	*G*	*10*	NVCC_SD2	1.65	1.8	1.95	M3	100	1	-	-

Note:

1. *The content in **italics** means that the power rails can support dual-range measurement.*
2. *Sensing resistance and range values correspond to values populated on 8MPLUSLPD4-PWR board revision A1. See the latest schematic for detailed information.*
3. *The Rs2 value is calculated by adding a low shunt resistor and a high shunt Rs1. See the latest schematic for detailed information.*

2.2 Power tree

Figure 1 shows the power tree of the PWR CPU board. The PWR board (8MPLUSLPD4-PWR) supply design is identical to the standard CPU board (8MPLUSLPD4-CPU) but with current sense resistors of a suitable value inserted between the PMIC and CPU for each key power rail measurement.

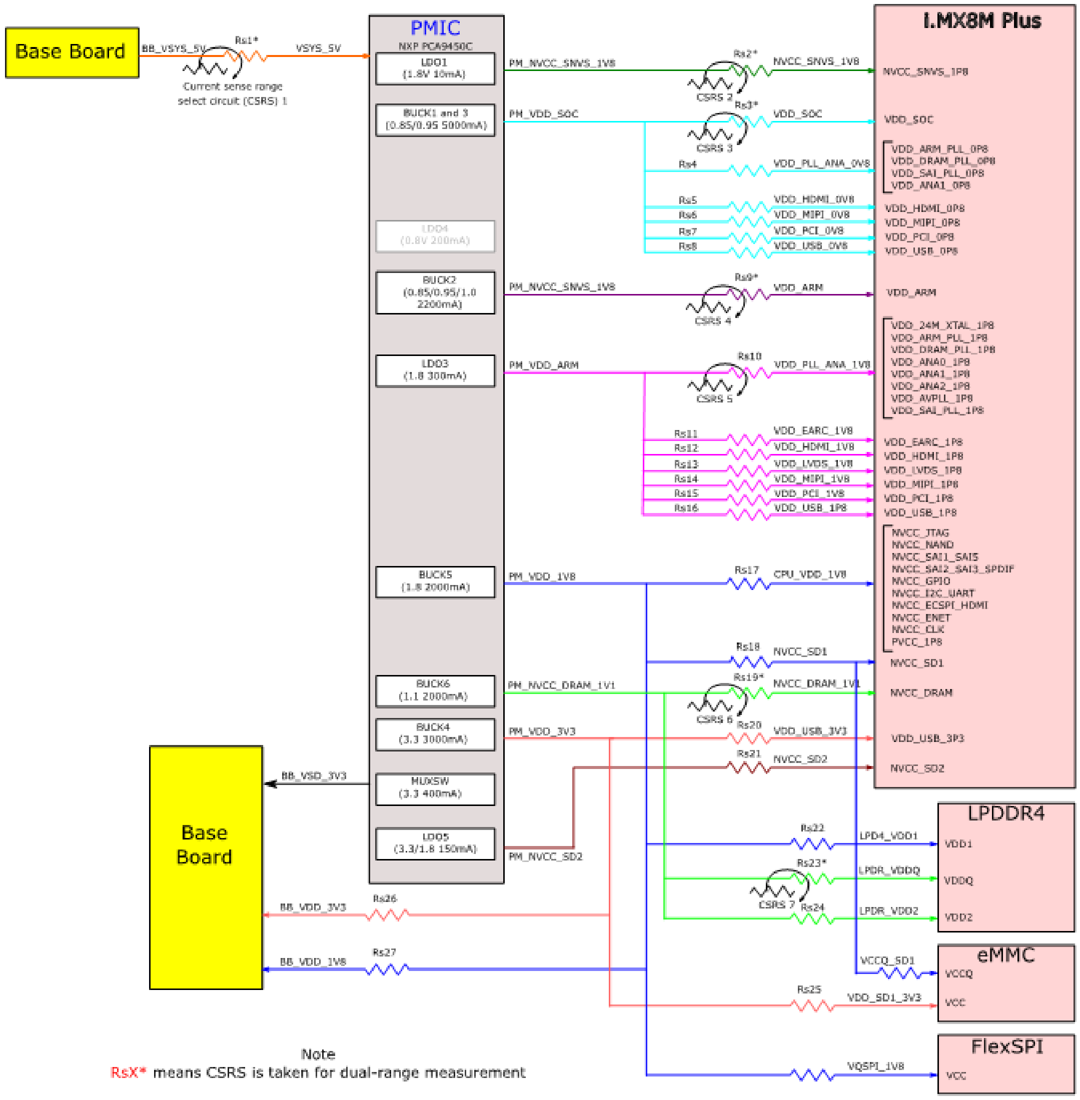


Figure 1. 8MPLUSLPD4-PWR board power tree

2.3 Power groups

There are four power groups defined on the 8MPLUSLPD4-PWR board:

- GROUP_SOC: i.MX 8MP SoC power consumption, including SOC PHY's contribution.
- GROUP_SOC_FULL : Full i.MX8MP SoC power consumption, including PHY's and IO's contribution.
- GROUP_DRAM: DRAM device power consumption.
- GROUP_PLATFORM: Full platform consumption. This power group covers the power consumption of all components supplied by the PMIC, including PMIC's own consumption.

Table 2 lists the specific power rails included in the groups. These power groups can be monitored by the supported power measurement applications.

Table 2. Power groups definition

Power group			Power rail
GROUP_PLATFORM (VSYS_5V rail)	**GROUP_SOC_FULL**	**GROUP_SOC**	NVCC_SNVS_1V8
			VDD_SOC
			VDD_ARM
			NVCC_DRAM_1V1
			VDD_PLL_ANA_0V8
			VDD_HDMI_0V8
			VDD_MIPI_0V8
			VDD_PCI_0V8
			VDD_USB_0V8
			VDD_PLL_ANA_1V8
			VDD_EARC_1V8
			VDD_HDMI_1V8
			VDD_LVDS_1V8
			VDD_MIPI_1V8
			VDD_PCI_1V8
			VDD_USB_1V8
			VDD_USB_3V3
		i.MX8MP IO power	CPU_VDD_1V8
			NVCC_SD1
			NVCC_SD2
	GROUP_DRAM		LPD4_VDD1
			LPD4_VDDQ
			LPD4_VDD2
	Other component’s power supplied by PMIC		BB_VDD_1V8
			BB_VDD_3V3

2.4 Board view

The overview of the 8MPLUSLPD4-PEVK platform is shown in Figure 2. It is fully software compatible with the 8MPLUSLPD4-EVK platform. The difference between the two platforms is only the addition of the power measurement function on the 8MPLUSLPD4-PEVK platform described in this document. To get more information about the common features of the board, see the *i.MX 8M Plus LPDDR4 EVK Board Hardware User's Guide* (document IMX8MPEVKHUG).

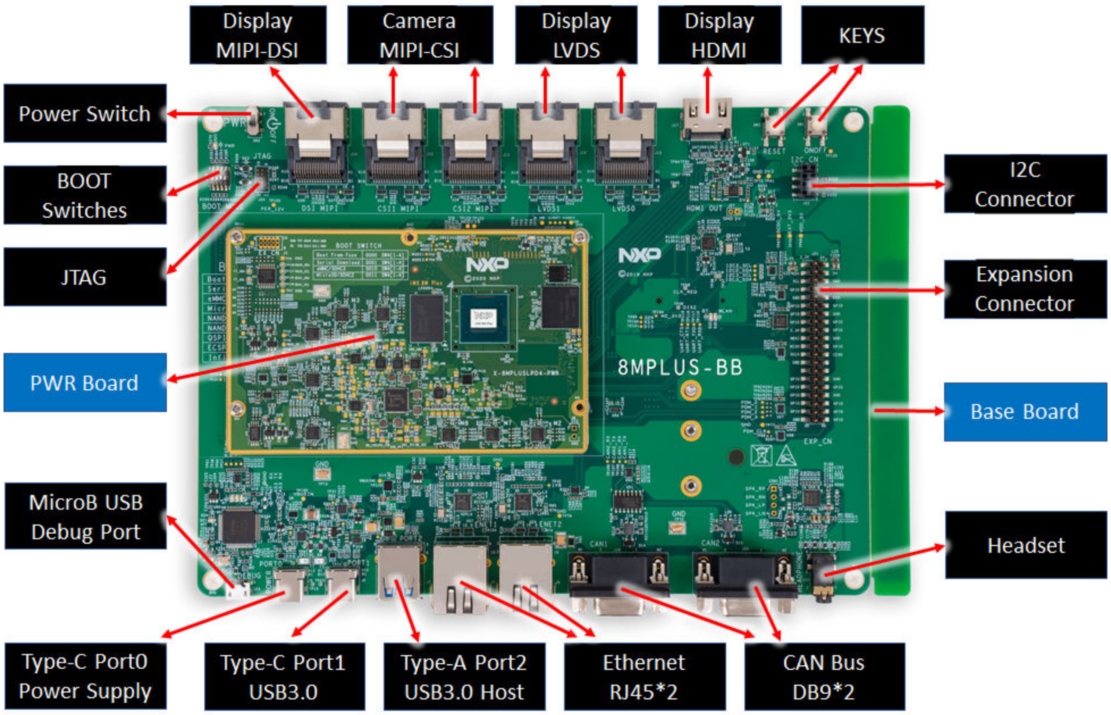


Figure 2. 8MPLUSLPD4-PEVK platform

Chapter 3
Getting started

3.1 Power measurement acquisition applications

The following two applications are available to acquire real-time power data from the board, both are publicly available on GitHub:

- BCU software: Command-line tool
- PMT software: GUI-based tool

The BCU software is a command-line tool, designed to control boards/platforms that support remote control and power measurement. It provides functions such as on/off key operation, board reset, setting boot mode, JTAG debug, and power measurement through the USB debug port.

The PMT software is a GUI-based tool. It offers functionalities similar to the BCU, but thanks to the GUI, it also allows real-time profiling of power rails (graphical monitoring of power, voltage, and current).

The sampling rate achieved on the BCU is higher than on the PMT due to the GUI processing on the latter.

- BCU is therefore provided for power acquisition where fine live monitoring is not the primary purpose, e.g. remote acquisition, thanks to its higher sampling rate.
- PMT is therefore provided for live power monitoring activities during system design, debug phases, and also post-processing analysis, thanks to its enhanced GUI.

Note that power acquisitions made using the BCU can be imported into the PMT for post-processing analysis to benefit from the advantages of the two applications, higher sampling resolution of the BCU, and enhanced visibility of power rail activities of the PMT.

This document only shows the power measurement functions for the 8MPLUSLPD4-PEVK platform. For more information on the other functions of BCU and PMT, see the BCU release note in *GitHub* and *i.MX Power Measurement Tool* (document AN13119).

3.2 EEPROM configuration and usage

3.2.1 Description

The 8MPLUSLPD4-PWR board embeds a 32-kb System ID EEPROM (U28), connected to the I2C debug interface. This EEPROM is used to store the following useful board information:

- Board ID and revision
- SoC ID and revision
- PMIC ID and revision
- Number of measurable power rails on the board
- Board serial number (user-defined)

The PMT and BCU power measurement acquisition tools can detect the type of the board connected by reading the EEPROM.

NOTE

If the connected board is used for the first time, ensure that the EEPROM is correctly programmed. EEPROMs are programmed during the board-manufacturing process. You can reset the configuration if an error occurs. The current manufacturing EEPROM configuration settings are listed in the "Data" column in Table 3 (it is subject to change in the case of 8MPLUSLPD4-PWR board revision updates).

3.2.2 EEPROM configuration

Table 3 shows all the useful information about the board stored in the EEPROM. The default manufacturing settings are indicated in the "Data" column. The EEPROM configuration settings can be updated using the BCU or PMT tools, e.g. to set a new board-specific serial number value (BOARD_SN).

Table 3. Board configuration data

<table>
<tr><th>Information</th><th>Address of EEPROM</th><th>Data[1]</th><th>Comments</th></tr>
<tr><td>BOARD_ID</td><td>0x1A</td><td>NXP i.MX8MP EVK PWR Board</td><td rowspan="2">These two information are very important.
Ensure that they are set properly in the PMT YAML configuration file.
In the BCU, provide the correct [-brev=] value in the command if you do not use the default revision value.
Note: A1 is the latest board release to be adapted to your board version.</td></tr>
<tr><td>BOARD_REV</td><td>0x1C</td><td>A1</td></tr>
<tr><td>SOC_ID</td><td>0x1D</td><td>i.MX8MP</td><td>This is related to the [-board=] option and it cannot be changed manually in the BCU.</td></tr>
<tr><td>SOC_REV</td><td>0x1E</td><td>A1</td><td>In the BCU, provide the correct [-srev=] value in the command if you do not use the default revision value.
Note: A1 is the latest SoC revision to be adapted to your SoC board revision.</td></tr>
<tr><td>PMIC_ID</td><td>0x1F</td><td>PCA9450CHN</td><td rowspan="3">These are related to the [-board=] option and they cannot be changed manually in the BCU.</td></tr>
<tr><td>PMIC_REV</td><td>0x20</td><td>N/A</td></tr>
<tr><td>NBR_PWR_RAILS</td><td>0x21</td><td>27</td></tr>
<tr><td>BOARD_SN</td><td>0x22</td><td>1</td><td>Board-specific serial number, user-defined (range: 1~65535).</td></tr>
</table>

1. *EEPROM manufacturing current default settings*

3.2.2.1 EEPROM configuration in PMT

To program the EEPROM with the PMT application, use the “EEPROM_Programmer_Tool” YAML file provided in the GitHub PMT repository.

Complete the fields as follows:

- BOARD_ID: NXP i.MX8MP EVK PWR board
- BOARD_REV: A1
- SOC_ID: i.MX8MP
- SOC_REV: A1
- PMIC_ID: PCA9450CHN
- PMIC_REV: NOT FOUND
- NBR_PWR_RAILS: 27
- BOARD_SN: 24

The board configuration information stored in EEPROM are shown in Table 3. Ensure that each information is correctly filled with the needed data and then use the PMT to flash the EEPROM as per the following instructions.

Flash the board configuration into the EEPROM as follows:

```
$ python3 main.py eeprom -m write -f docs/EEPROM_Programmer_Tool.yaml
```

```
** Info collected.

/!\ You are going to overwrite EEPROM content, want to continue? Y/y/N/n
Y
I2C EEPROM - Which board are you using ? ( (1) imx8dxlevk / (2) imx8mpevkpwr(a0 / a1) )
2
** Writing to I2C EEPROM ...
** Done.
```

Read the EEPROM content as follows:

```
$ python3 main.py eeprom -m read
```

```
I2C EEPROM - Which board are you using ? ( (1) imx8dxlevk / (2) imx8mpevkpwr(a0 / a1) )
2
** Reading I2C EEPROM ...

CONFIG_FLAG: Programmed EEPROM
BOARD_ID: NXP i.MX8MP EVK PWR Board
BOARD_REV: A1
SOC_ID: i.MX8MP
SOC_REV: A1
PMIC_ID: PCA9450CHN
PMIC_REV: Unknown
NBR_PWR_RAILS: 27
BOARD_SN: 24

** Done.
```

3.2.2.2 EEPROM configuration in BCU

For a full description of the BCU tool, see the BCU release note in GitHub.

Flash the board configuration into the EEPROM as follows:

```
$ sudo ./bcu eeprom -w -board=imx8mpevkpwra1 -sn=24 -brev=A1 -srev=A1
```

The [-board=] option is used to set the BOARD_ID parameter. It must be set to "imx8mpevkpwra0" or "imx8mpevkpwra1" for the 8MPLUSLPD4-PWR board (mandatory). The "imx8mpevkpwra1" version is the latest board version at the moment of writing this document.

The [-brev=] and [-srev=] options are usually not used, because the [-board=BOARDNAME] option can specify a unique data template for writing to the EEPROM.

If the [-brev=] option is used, the board revision information is modified. If this option is not used, the default board revision is used. Use capital letters. If the [-srev=] option is used, the revision information of the SoC is modified. If this option is not used, the default SoC revision is used. Use capital letters.

If the [-sn=] option (range: 1~65535) is used, the board serial number is modified. If this option is not used, it is set to 1.

```
[redacted]@OptiPlex:~/work/bcu/build$ sudo ./bcu eeprom -w -board=imx8mpevkpwra1 -sn=24
version bcu_1.1.24-4-ge0633ae
board model is imx8mpevkpwra1
eeprom user SN will be set to 24

>>>>>> Registered AT24Cxx EEPROM on board >>>>>>
Write imx8mpevkpwra1 default values to FTDI EEPROM successfully

Board Info: NXP i.MX8MP EVK PWR Board Rev A1
  SoC Info: i.MX8MP Rev A1
 PMIC Info: PCA9450CHN
Number of available power rails: 27
Serial Number: 24

done
```

Read the EEPROM content. If the EEPROM on the board is flashed, use the following command to read it manually:

```
$ sudo ./bcu eeprom -r -board=imx8mpevkpwra1
```

```
[redacted]@OptiPlex:~/work/bcu/build$ sudo ./bcu eeprom -r -board=imx8mpevkpwra1
version bcu_1.1.24-4-ge0633ae
board model is imx8mpevkpwra1

>>>>>> Registered AT24Cxx EEPROM on board >>>>>>

Board Info: NXP i.MX8MP EVK PWR Board Rev A1
  SoC Info: i.MX8MP Rev A1
 PMIC Info: PCA9450CHN
Number of available power rails: 27
Serial Number: 24

done
```

3.3 Power acquisition using BCU

3.3.1 Download BCU binaries

If do not have the Snap Package installed, download and install the BCU binaries at GitHub. Figure 3 shows the file list.

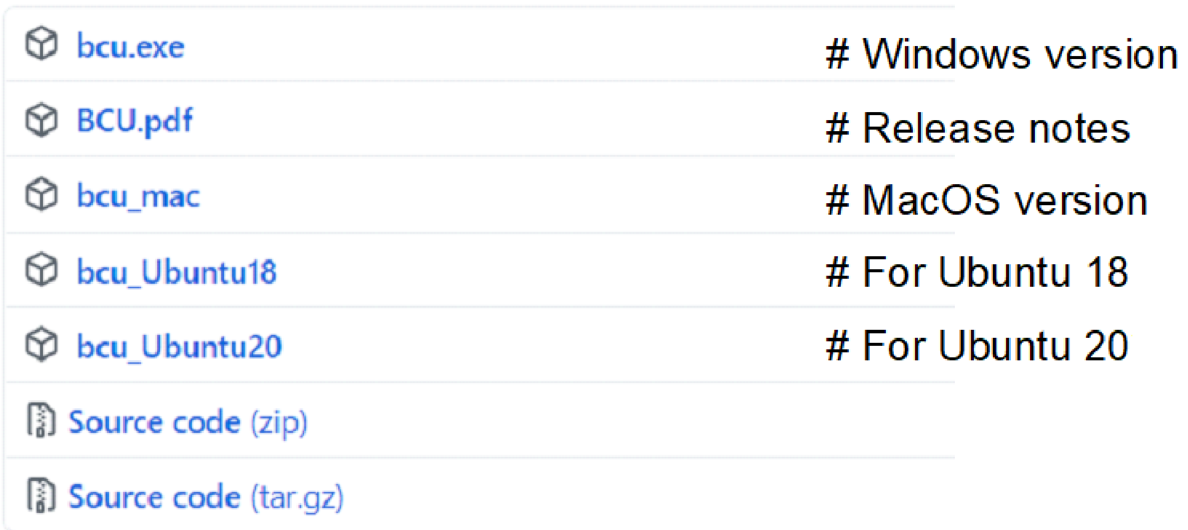


Figure 3. GitHub file list

For Ubuntu 18 and 20, the BCU Snap Package has been released in the Snap Store. Installing the BCU from the Snap Store is recommended. These binaries released on the GitHub release page do not support Ubuntu 16 anymore.

3.3.2 BCU running environment

The BCU supports Linux, Windows 10, and MacOS 11.x.

3.3.2.1 Linux distributions (x64)

If your Linux distribution supports Snap Store, install the BCU using below commands. There is no need to install other libraries:

```
$ sudo snap install --devmode bcu-nxp; sudo snap alias bcu-nxp bcu
```

If you are using Ubuntu 18.04 (or above) and want to directly use the binaries released on the BCU GitHub release page, install these libraries using the following command:

```
$ sudo apt-get install libftdi1-dev libyaml-dev openssl pkgconf libncurses5-dev libssl-dev libconfuse-
dev
```

3.3.2.2 Windows 10 (x64)

The BCU only provides the Windows 10 binary on the GitHub release page.

- Install the FTDI Driver for Windows 10.
- If using Windows 10 version below 1.2.0, install the vs2017 redistribute package.

3.3.2.3 MacOS 11.x (x64 and arm64)

The BCU only provides the MacOS binary on the GitHub release page.

- MacOS 11.x Big Sur (or above, x64):
  - Install Homebrew:

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/
install.sh)"
```

— Then install the other libraries by Homebrew:

```
$ brew install libftdi libyaml
```

— The binary is only tested on MacOS 11.1. It may work on previous versions of MacOS, but it is not guaranteed.

- MacOS 11.x Big Sur (or above, arm64):

— The Apple Silicon can run the x64 binary using Rosetta 2, but all libraries should be installed to the same path as for x64:

```
$ arch -x86_64 $SHELL
```

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/
install.sh)"
```

— Install the other libraries by Homebrew to the default path(`/usr/local/Cellar`):

```
$ brew install libftdi libyaml
```

3.3.3 Typical operations

3.3.3.1 Connection

Connect the Micro-B USB Debug Port to a PC, and connect the Type-C USB Port0 to the power supply. Then power on the board using the power switch.

3.3.3.2 Real-time monitor

Open a terminal or a command prompt/PowerShell on the PC and change the directory to the location of the BCU files. Then use the command below to monitor the power consumption at a specified page refresh frequency using the [-hz=value] option (for example, [-hz=1] is one time per second).

For Linux OS and macOS:

```
$ sudo ./bcu monitor -hz=1 -board=imx8mpevkpwra1
```

For Windows OS:

```
.\bcu.exe monitor -hz=1 -board=imx8mpevkpwra1
```

A TUI appears on the screen (Figure 4).

In this interface, you can change some configurations using the following hot keys:

- Left-click anywhere to freeze the display and press any key to recover refreshing.
- Press the following hot keys for typical operations:

  ‘1’ = Reset avg calculation

  ‘2’ = Reset max and min calculation

  ‘3’ = Reset avg, max, and min calculation

  ‘4’ = Switch current unit to show: auto/mA/uA

  ‘5’ = Reset the board

  — The TUI interface asks for the boot mode selection afterwards:

```
Available boot modes:
 0 fuse
 1 usb
```

```
2 emmc
3 sd
4 nand_256
5 nand_512
6 qspi_3b_read
7 qspi_hyperflash
8 ecspi
9 reserved
others boot from BOOT SWITCH
```

'6' = resume the board (function reserved)

'Ctrl+C' = exit.

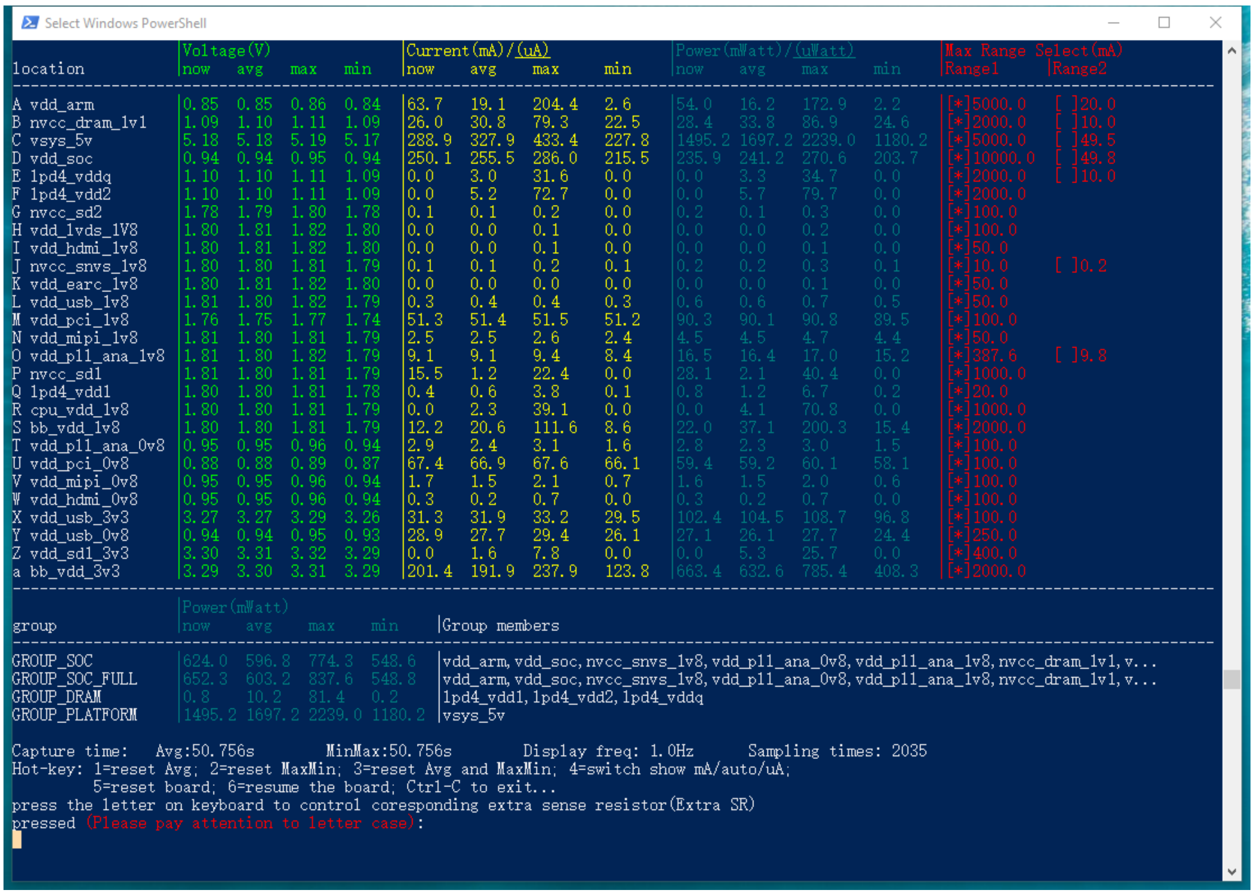


Figure 4. Power monitor display

3.3.3.3 Board reset operation

The BCU reset command sets all rails to the default large current measurement range. This ensures that the selected current-sensing resistor does not lower the voltage of a power rail below the operating range of the device and cause failure to resume proper system operation.

The reset command is as follows:

- For Linux: `$ sudo ./bcu reset -board=imx8mpevkpwra1`
- For Windows: `.\bcu.exe reset -board=imx8mpevkpwra1`
- Hotkey in TUI: 5

3.3.3.4 Data export

Use the following BCU command-line options to export data:

- `[-dump]` outputs the power data to the terminal in the CSV format. The file is named *monitor_record.csv*.
- `[-dump=filename]` outputs the data and names the file as *filename.csv*.

The interval of data sampling while using the data export command is related to the computer performance, system version, and number of sampled power rails.

For Windows OS, it also relates to the refresh settings. You can set the refresh frequency to less than 0.5 Hz using the [-hz=value] option and disable the TUI display using the [-nodisplay] option to achieve a higher sample rate.

For Linux OS, the refresh settings have no impact on the sample rate.

The BCU can achieve the following minimum sampling interval:

- 20 ms/sample (50 samples/s) for all 27 power rails at the same time
- 4 ms/sample (250 samples/s) for 4 rails on one PAC1934 monitor (M1 to M8)

See Table 1 for the corresponding power rails for monitors M1 to M8. See Other configurations for how to reduce the exported power rails with the “show_id” parameter.

The export data command examples are as follows:

- For Linux OS: `$ sudo ./bcu monitor -dump=filename -board=imx8mpevkpwra1`
- For Windows OS: `.\bcu.exe monitor -nodisplay -hz='0.5' -dump=filename -board=imx8mpevkpwra1`

3.3.3.5 Sense resistance reconfiguration

The BCU software supports the customer configuration changes to the physical measurement hardware. The sensing resistors may have to be changed on the PWR board to achieve the best measurement for your specific testing. The BCU software provides a simple method to define the new sensing resistors' values (Rsense) for a user-defined measurement range.

PAC1934 supports a 100-mV full-scale voltage-sense range with 16-bit resolution. Thus, the full-scale current can be calculated as follows:

Measurement range (mA) = 100mV / Rsense (ohm)

The BCU software configurations are stored in the *config.yaml* file, which is in the BCU software install folder. This file is created by the BCU software upon the first execution of the BCU. Open the *config.yaml* file and find the text beginning with “boardname: imx8mpevkpwra1”, as follows:

```
boardname: imx8mpevkpwra1
mappings:
 - vdd_arm : { rsense1: 20 , rsense2: 5010 , show_id: 1 }
#Range: rsense1=>5000.0mA, rsense2=>20.0mA
 - nvcc_dram_1v1 : { rsense1: 50 , rsense2: 2050 , show_id: 2 }
#Range: rsense1=>2000.0mA, rsense2=>48.8mA
 - vsys_5v : { rsense1: 20 , rsense2: 420 , show_id: 3 }
#Range: rsense1=>5000.0mA, rsense2=>238.0mA
- vdd_soc : { rsense1: 10 , rsense2: 1010 , show_id: 4 }
 #Range: rsense1=>10000.0mA, rsense2=>100mA
…
…
```

When a physical sense resistor is changed, the "rsense1" or "rsense2" values (in milliOhms) can be updated within the *config.yaml* file to match the new resistor value.

ATTENTION

Do not modify any spaces within the *config.yaml* file.

3.3.3.6 Other configurations

1. "show_id" parameter

   The “show_id” parameter within the *config.yaml* configuration file can set the display order of the power rails. If the “show_id” parameter is set to 0, then this rail will not be sampled. It will not be displayed in the TUI and dumped into the CSV file. Restart the BCU to bring the updated *config.yaml* file into effect. The following is a “show_id” parameter configuration example:

```
boardname: imx8mpevkpwra1
mappings:
 - vdd_arm : { rsense1: 20 , rsense2: 5010 , show_id: 3 }
#Range: rsense1=>5000.0mA, rsense2=>20.0mA
 - nvcc_dram_1v1 : { rsense1: 50 , rsense2: 2050 , show_id: 2 }
#Range: rsense1=>2000.0mA, rsense2=>48.8mA
 - vsys_5v : { rsense1: 20 , rsense2: 420 , show_id: 1 }
#Range: rsense1=>5000.0mA, rsense2=>238.0mA
 - vdd_soc : { rsense1: 10 , rsense2: 1010 , show_id: 0 }
#Range: rsense1=>10000.0mA, rsense2=>100.0mA
```

   In this example, the display order of "VDD_ARM" and "VSYS_5V" changed and the "VDD_SOC" data will not be sampled and displayed, as shown in Figure 5.

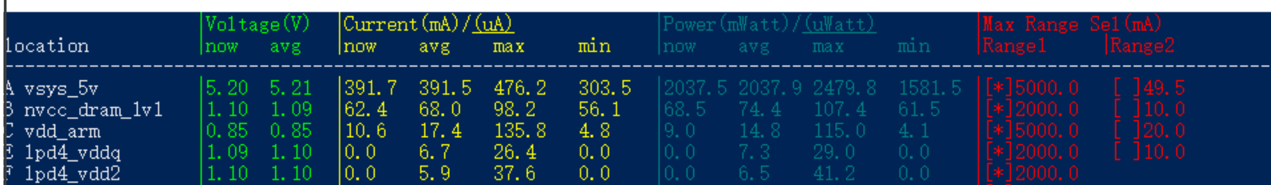


Figure 5. “show_id” parameter configuration example

2. Dual measurement range

   The TUI interface provides direct user control over the power rails with the dual measurement range. Enter the power rail location value to switch the sense resistor to measure the corresponding power rail within a different current range. For example, press “A” to switch the current range for “VDD_ARM” from 5000 mA to 20 mA. Press “A” again to switch back to 5000 mA. For the i.MX8MPLUS PWR board, there are 7 power rails that support the dual measurement range.

Table 4. Power rails

Location	Power Rail	Range1 (mA)	Rs1 (OHM)	Range2 (mA)	Rs2 (OHM)
A	VDD_ARM	5000	0.02	20	5.01
B	NVCC_DRAM_1V1	2000	0.05	48.8	2.05
C	VSYS_5V	5000	0.02	238	0.42
D	VDD_SOC	10000	0.01	100	1.01
E	LPD4_VDDQ	2000	0.05	48.8	2.05
J	NVCC_SNVS_1V8	10	10	0.2	509
O	VDD_PLL_ANA_1V8	400	1.008	9.1	11

CAUTION

This operation physically switches the sense resistance on the power path. The measured current must not exceed the measurement range, otherwise the voltage drop across the sense resistor may lower the supply voltage below the operating range. This may cause a device operation failure. For example, if using the reset button on the board or a Linux OS command to reset from a low-power mode, set all rails to the large measure range for power safety. If using the BCU remote control command to reset the board according to Board reset operation (recommended), all dual-range rails are set automatically.

3.4 Power acquisition using PMT

For detailed instructions on how to set up and run the PMT on the MX8MPLUSLPD4-PEVK platform, see the README file in the PMT GitHub or the *i.MX Power Measurement Tool* (document AN13119).

The PMT GUI running on the i.MX8MPLUS PWR board is shown in Figure 6.

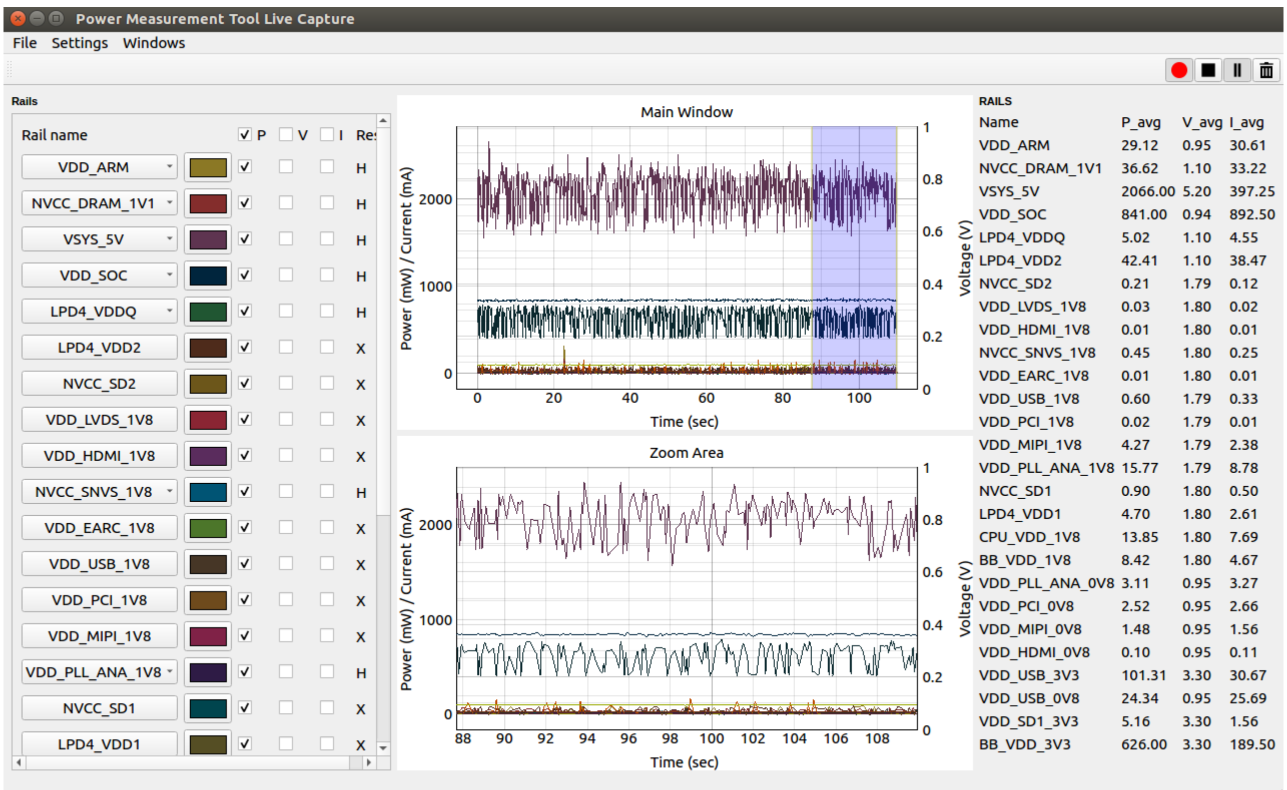
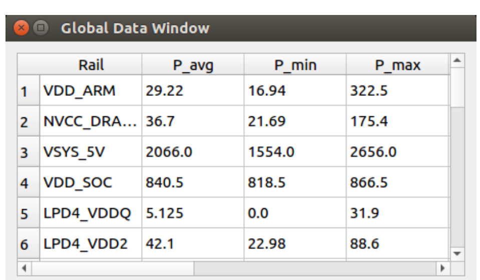


Figure 6. PMT monitoring main window display

Additional windows can be displayed in the "Windows" menu, showing the following:

- Power measurement information since the start of the capture:


Global Data Window

	Rail	P_avg	P_min	P_max
1	VDD_ARM	29.22	16.94	322.5
2	NVCC_DRA...	36.7	21.69	175.4
3	VSYS_5V	2066.0	1554.0	2656.0
4	VDD_SOC	840.5	818.5	866.5
5	LPD4_VDDQ	5.125	0.0	31.9
6	LPD4_VDD2	42.1	22.98	88.6

- Power measurement information data in the zoom region (in the blue color):

Zoom Data Window

Time : 2.0 sec

	Rail	P_avg	P_min	P_max	V_avg	V_min
1	VDD_ARM	40.8	19.73	87.44	0.9517	0.9463
2	NVCC_DRA...	35.75	23.47	66.5	1.103	1.1
3	VSYS_5V	2124.0	1666.0	2432.0	5.2	5.19
4	VDD_SOC	842.0	830.5	857.0	0.9424	0.9375
5	LPD4_VDDQ	9.2	0.0	29.14	1.102	1.095

- Power measurement information at the mouse pointer location:

Mouse Pointer Data Window

Time : 3.013826748544652 sec

	Rail	Power (mW)	Voltage (V)	Current (mA
1	VDD_ARM	21.11	0.9478	22.28
2	NVCC_DRA...	28.23	1.1	25.67
3	VSYS_5V	1860.0	5.195	358.0
4	VDD_SOC	841.5	0.942	893.5
5	LPD4_VDDQ	0.0	1.102	0.0

Chapter 4
Revision history

Table 5. Revision history

Revision number	Date	Substantive changes
0	29 April 2021	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 29 April 2021
Document identifier: IMX8MPPBUG