KA49703A Reference Platform
PC GUI Software User Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation Japan and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing document only for reference purposes of KA49703A PC GUI Software based system design. Nuvoton assumes no responsibility for errors or omissions.
All data and specifications are subject to change without notice.

For additional information or question, please contact Nuvoton Technology Corporation Japan.
www.nuvoton.co.jp

TABLE OF CONTENTS

1. OVERVIEW OF THE KA49703A PC GUI SOFTWARE 3
1.1 Overview 3
2. CONNECTION SETUP 4
2.1 Components 4
2.2 Hardware Configuration 5
2.2.1 BMIC Standalone Configuration 6
2.2.2 BMIC with PM-IC Configuration 7
2.2.3 BMIC and PM-IC Configuration with COMM-IC 9
2.3 Hardware Connection for PM-IC Board Operation 10
2.3.1 Connection for Current Measurement 10
2.3.2 Connection for Voltage Measurement 10
2.3.3 Connection for Insulation Detection function 11
2.3.4 Connection for VBAT/Relay Diagnosis function 12
2.4 Power Up the Boards 13
2.5 Firmware Update 14
3. PC GUI SOFTWARE DESCRIPTION 15
3.1 Configuration Window 15
3.2 Main Window 16
3.3 Sub Window 30
3.3.1 Functions – Register R/W 30
3.3.2 Functions – Continuous R/W 32
3.3.3 Functions – Cell Balance 33
3.3.4 Functions – Diagnostic 36
3.3.5 Functions – Fast CV Log 37
3.3.6 Functions – Graphs 39
3.4 PM-IC Window 42
3.5 Sub Window (PM-IC Functions) 42
3.5.1 Functions – Current Measurement 45
3.5.2 Functions – PM-IC Graphs 46
3.5.3 Functions – Insulation Detection 47
3.5.4 Functions – Coulomb Counter 50
3.5.5 Functions – VBAT/Relay Diagnosis 52
3.5.6 Functions – VADC 53
4. LED ALLOCATION AND INDICATION 54
4.1 LED Definition for Operation Mode and Error Factor 54
4.2 LED Definition for Firmware Update Operation 55
REVISION HISTORY 56

1. OVERVIEW OF THE KA49703A PC GUI SOFTWARE

1.1 Overview

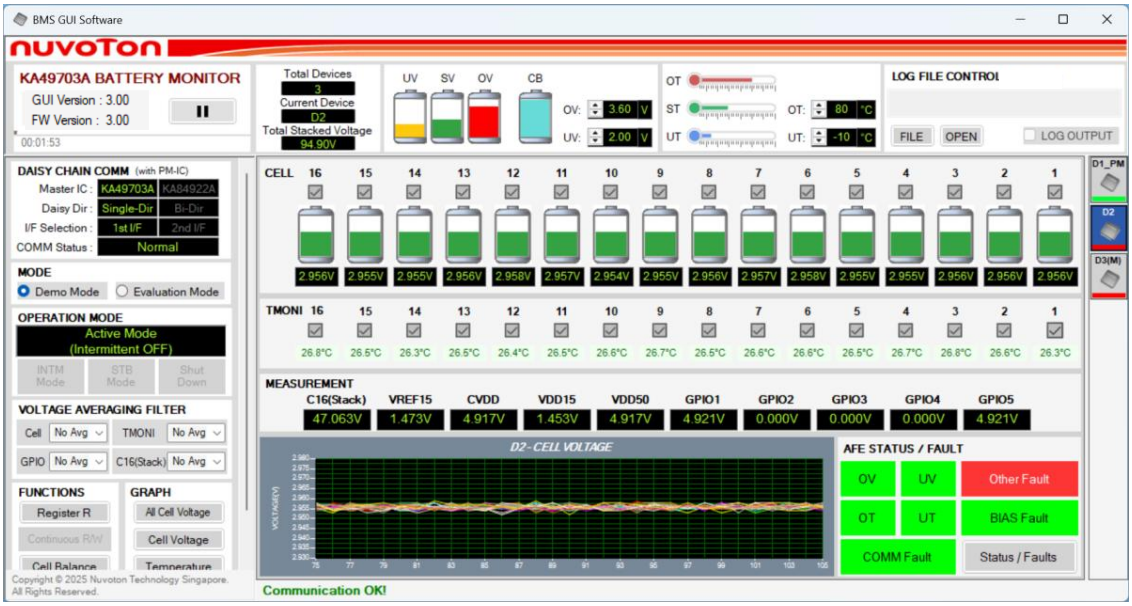


Figure 1.1 KA49703A PC GUI Software Main Window

The PC GUI software for the KA49703A Reference Software Platform is tailored to interface with the Nuvoton Battery Monitoring IC (KA49703A) through the Nuvoton MCU (M483SGCAE) Board, featuring dedicated firmware integration. When the software launches, it automatically verifies firmware compatibility with the PC GUI version and offers the option to perform a firmware update if applicable.

The MCU board houses the essential firmware components of the BMIC, including device drivers, middleware, and demonstration applications. These core firmware elements empower the PC GUI to execute its high-level functions.

This Manual provides a comprehensive overview of the PC GUI Software's functional operations. For detailed information on the MCU Board Firmware, please refer to "KA49703A Reference Platform Software User Manual".

The PC GUI software establishes communication with the MCU via a USB interface, enabling it to issue commands to the BMIC. The MCU, in turn, communicates with the BMIC through an SPI interface. The software retrieves data from the MCU at 1 second intervals and can also control BMIC registers upon request. Importantly, this PC GUI software is fully compatible with Windows 11.

The software provides a comprehensive set of features. In addition to firmware update capability, it supports daisy-chained BMS configuration, display of both raw and converted BMIC register data, real-time register monitoring, data logging, graphical data visualization, and full BMIC register map control for all devices in the daisy chain. Furthermore, the PC GUI offers two primary operating modes: Demo Mode and Evaluation Mode. Users can also manage operation modes and perform demonstrations of protection, cell balancing, diagnostic and voltage averaging filter functions.

The PC GUI software introduces battery pack monitoring via the KA84917A pack monitoring IC. It is compatible with the PM-IC board and supports current measurement, insulation detection, battery pack voltage measurement, and relay diagnostic.

Note: *The PC GUI Software has been tested with Microsoft Windows 11. Its compatibility with earlier version of Windows has not been fully evaluated and, therefore, cannot be guaranteed.*

2. CONNECTION SETUP

2.1 Components

To demonstrate the KA49703A Reference Platform using PC GUI, the following items are required for user setup:

1) MCU controller board (Main device : MCU M483SGCAE)
2) Analog Front End (AFE) board (Main device : BM IC KA49703A)
3) Pack Monitoring IC board (Main device : PM IC KA84917UA) (Optional)
4) Communication IC board (Main device : COMM-IC KA84922UA) (Optional)
5) Jumper wire (for Daisy Chain communication)
6) USB Type A to Type C cable
7) Nu-Link Pro Programmer (Optional: For Software Development & Firmware flash)

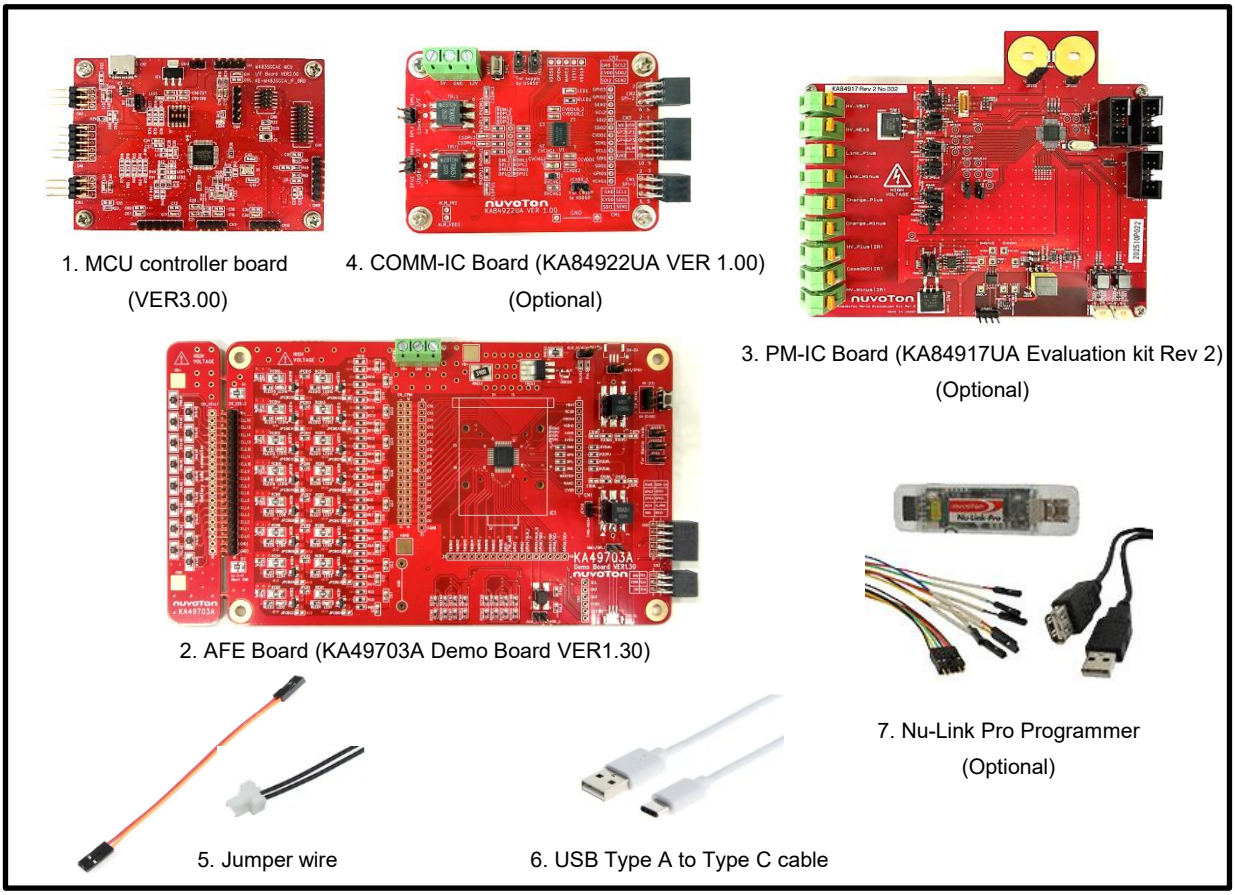


Figure 2.1 Components of KA49703A Reference Platform

2.2 Hardware Configuration

Connect the hardware as shown below.

The BMS daisy-chain setup supports multiple hardware configurations, including:

- Battery Monitoring IC (KA49703A) only
- Battery Monitoring IC (KA49703A) with Pack Monitoring IC (KA84917A)
- Battery Monitoring IC (KA49703A) with Communication IC (KA84922A)
- Battery Monitoring IC (KA49703A) with both Pack Monitoring IC (KA84917A) and Communication IC (KA84922A)

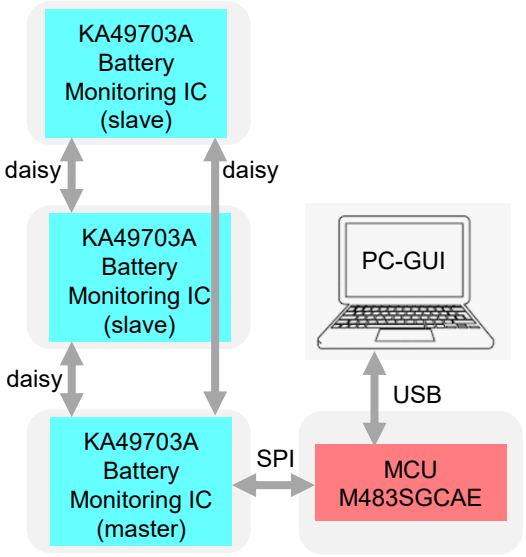


Figure 2.2 Hardware Configuration – BMIC only

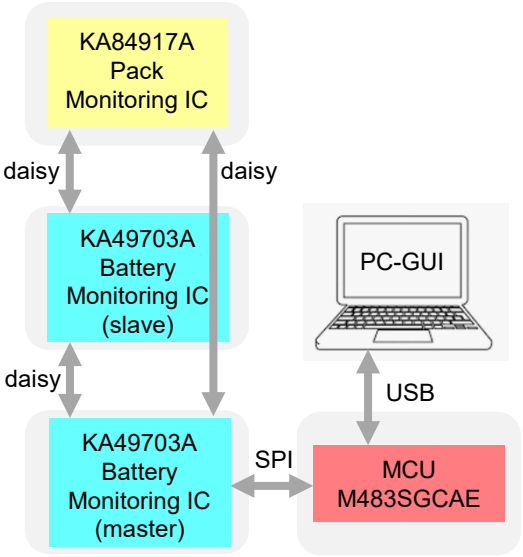


Figure 2.3 Hardware Configuration – BMIC with PM-IC

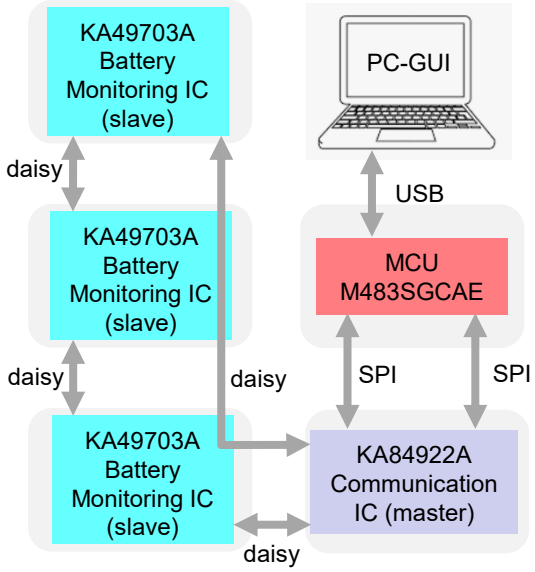


Figure 2.4 Hardware Configuration – BMIC + COMM IC

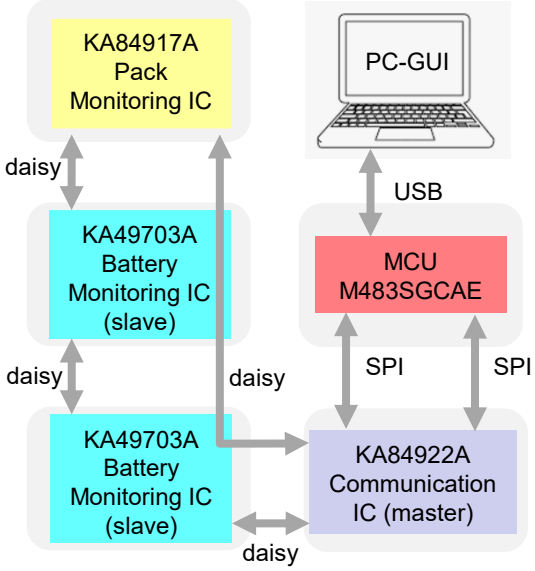


Figure 2.5 Hardware Configuration – BMIC + COMM IC + PM-IC

2.2.1 BMIC Standalone Configuration

The setup of BMIC standalone configuration:

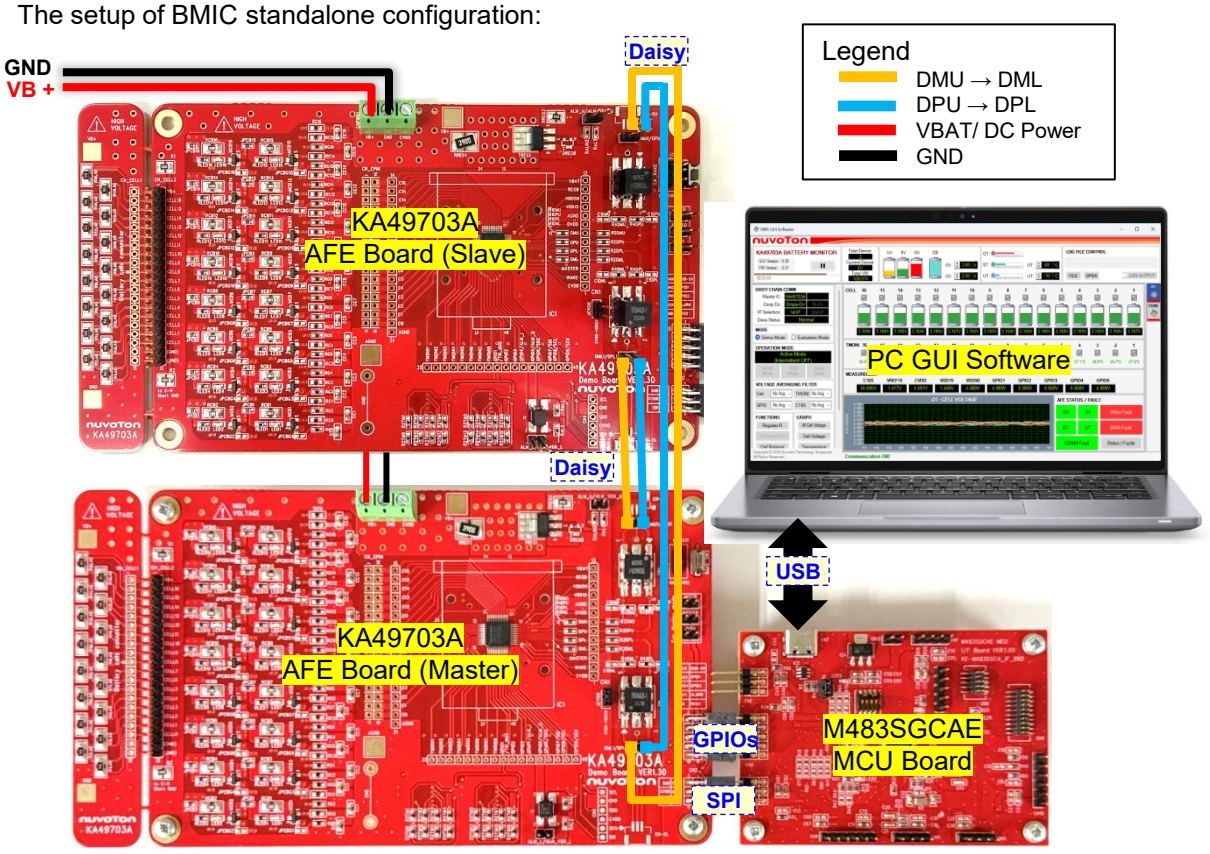


Figure 2.6 Setup of BMIC Standalone Configuration

Below are detailed port/jumper connections of AFE Board / MCU Board, ensure all parts (jumper, connectors) are firmly attached and oriented correctly.

Power supply	• AFE board: 12.5V ~ 76.8V
Daisy Connection	1st Interface: DMU (Master) ⇔ DML (Slave) DPU (Master) ⇔ DPL (Slave)
	2nd Interface: DMU (Slave) ⇔ DML (Master) (loop back) DPU (Slave) ⇔ DPL (Master)
SPI/GPIO Connector	CN3 (Master Board) ⇔ CN5 (MCU Board) CN2 (Master Board) ⇔ CN3 (MCU Board)
Jumper	▪ AFE Master: JMSTR short, JCVDD open ▪ AFE Slave: JMSTR open, JCVDD short ▪ MCU Board: J2 short
USB Port	CN7 (Type-C USB port of MCU board) ⇔ PC USB port

Table 2.1 Detailed connections for BMIC standalone configurations

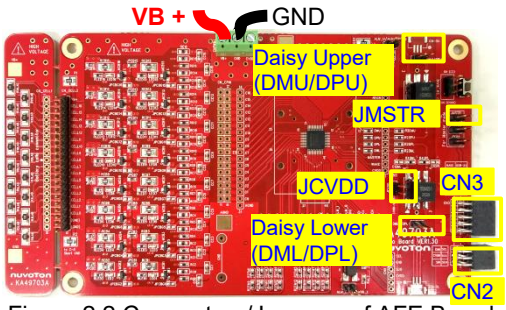


Figure 2.2 Connectors/Jumpers of AFE Board

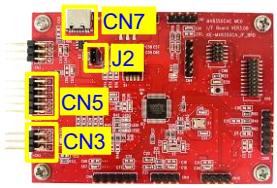


Figure 2.7 Connectors/Jumpers of MCU Board

2.2.2 BMIC with PM-IC Configuration

The setup of BMIC with PM-IC configuration:

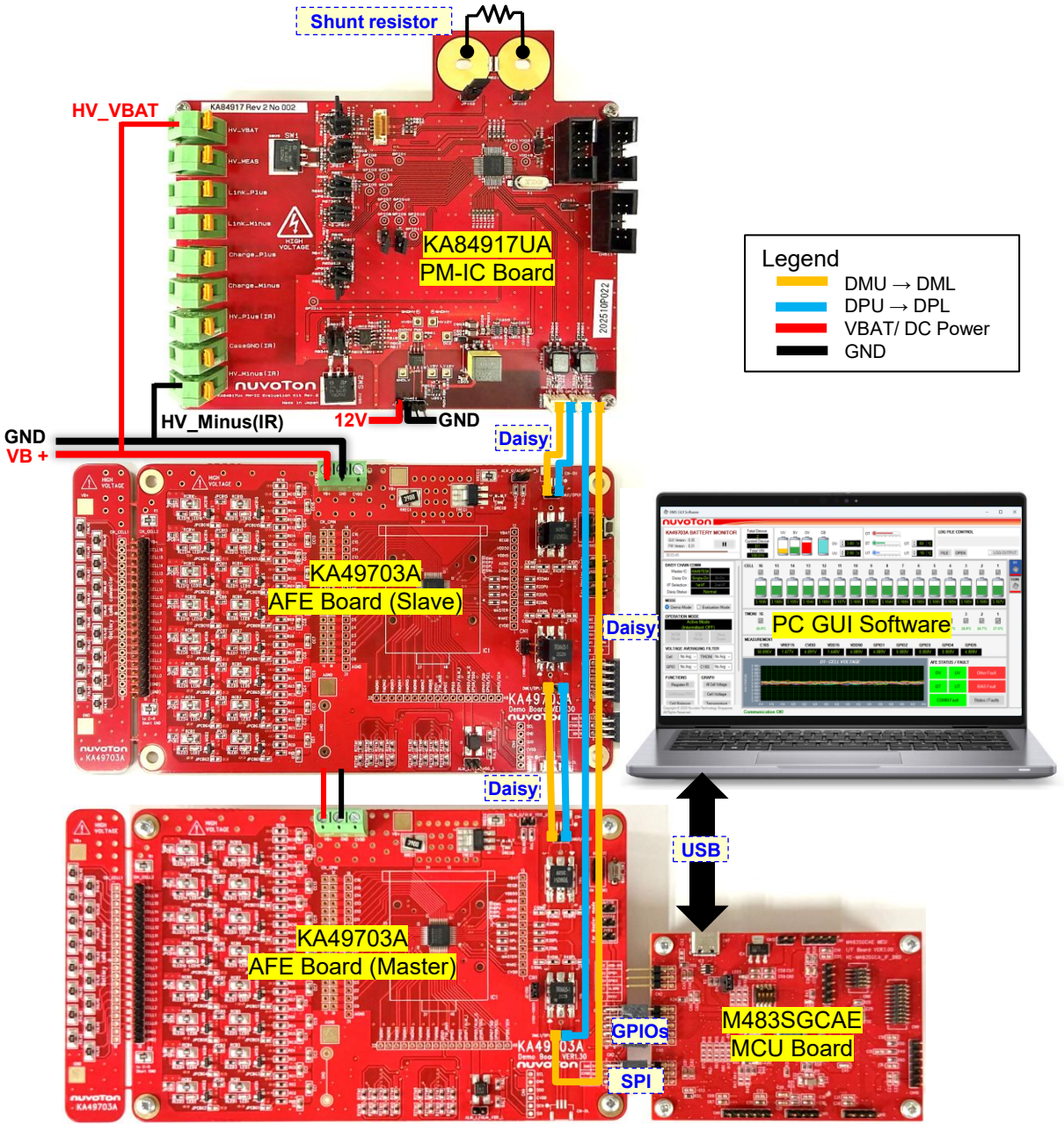


Figure 2.8 Setup of BMIC with PM-IC Configuration

Daisy Connection	1st Interface: DMU (AFE Master) ⇔ DML (AFE Slave) DPU (AFE Master) ⇔ DPL (AFE Slave) DMU (AFE Slave) ⇔ DML (CN302 of PM-IC board) DMU (AFE Slave) ⇔ DPL (CN302 of PM-IC board)
	2nd Interface: DMU (CN301 of PM-IC board) ⇔ DML (Master) (loop back) DPU (CN301 of PM-IC board) ⇔ DPL (Master)

Table 2.2 Detailed daisy connections for BMIC with PM-IC configurations

Below are detailed port/jumper connections of PM-IC Board, ensure all parts (jumper, connectors) are firmly attached and oriented correctly.

Power supply	• PM-IC board power supply: 12V on CN401 pin4(+) / CN401 pin3(-) of PM-IC Board • VBAT (battery pack supply): up to 72V/600V/1200V for HV_VBAT/ HV_Minus (IR)			
Jumper	▪ JP903 short – pull up GPIO8 for SW1 control (for Relay Diagnosis Function) ▪ JP902 short – pull up GPIO9 for SW2 control (for Insulation Detection Function)			
Connection (for Current Measurement)	Mount Shunt Resistor on R621 • Connect the positive side of the shunt resistor to pin JP102(1) and the negative side of the shunt resistor to pin JP103(2).			
	OR using DC Supply Without Shunt Resistor: • Connect +/− to JP102(1)/JP103(2), and short JP103(2) to JP103(1) GND. (Alternatively, connect +/− to JP103(2)/JP102(1) and short JP102(1) to JP102(2) GND			
Connection (for Voltage Measurement)	**Terminals**	**48V mode**	**400V mode**	**800V mode**
	HV_VBAT	JP open	JP802 short	JP802, JP801 short
	HV_MEAS	JP open	JP815 short	JP815, JP814 short
	Link_Plus	JP open	JP813 short	JP813, JP812 short
	Link_Minus	JP open	JP811 short	JP811, JP810 short
	Charge_Plus	JP open	JP807 short	JP807, JP806 short
	Charge_Minus	JP open	JP809 short	JP809, JP808 short
	HV_Plus (IR)	JP open	JP805 short	JP805, JP804 short

Table 2.3 Detailed connections for BMIC with PM-IC configurations

⚠ Warning:

The PM-IC VBAT input can be connected to voltages up to 1200 V. Exercise extreme caution when handling high-voltage connections to avoid electric shock.

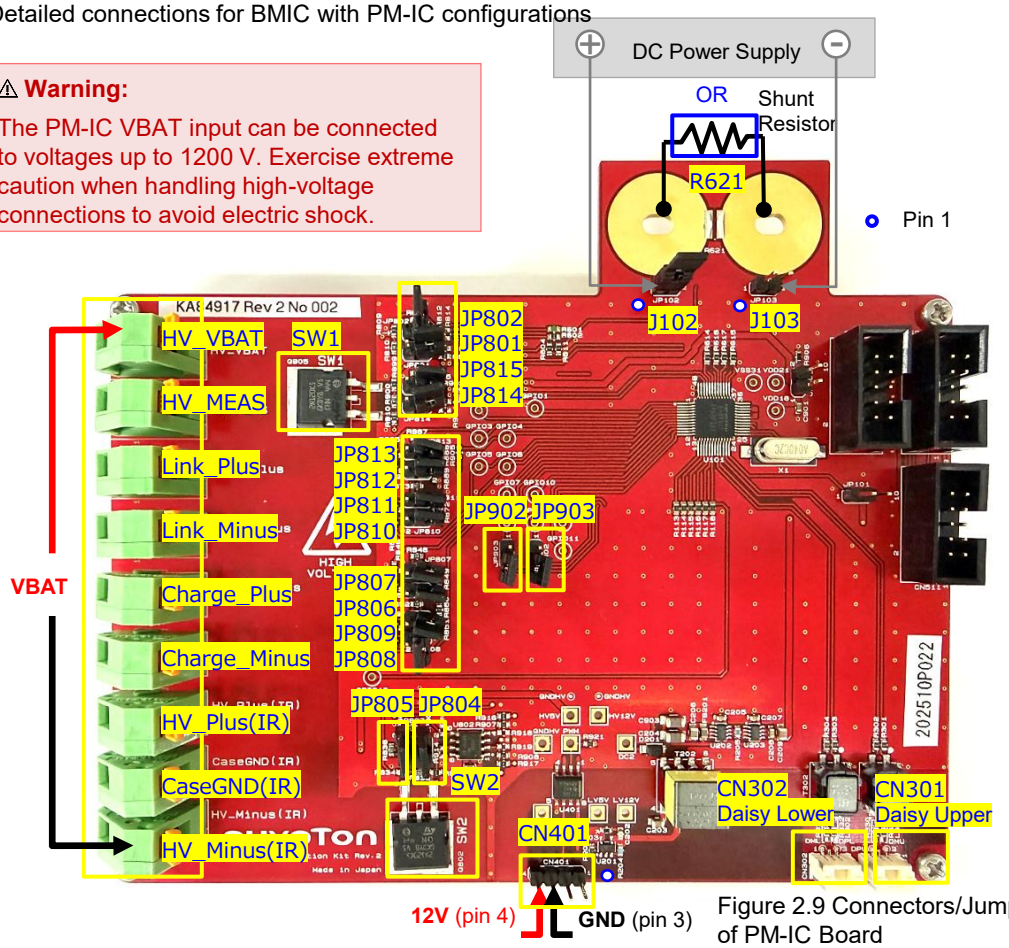


Figure 2.9 Connectors/Jumpers of PM-IC Board

2.2.3 BMIC and PM-IC Configuration with COMM-IC

The setup of BMIC and PM-IC configuration with COMM-IC:

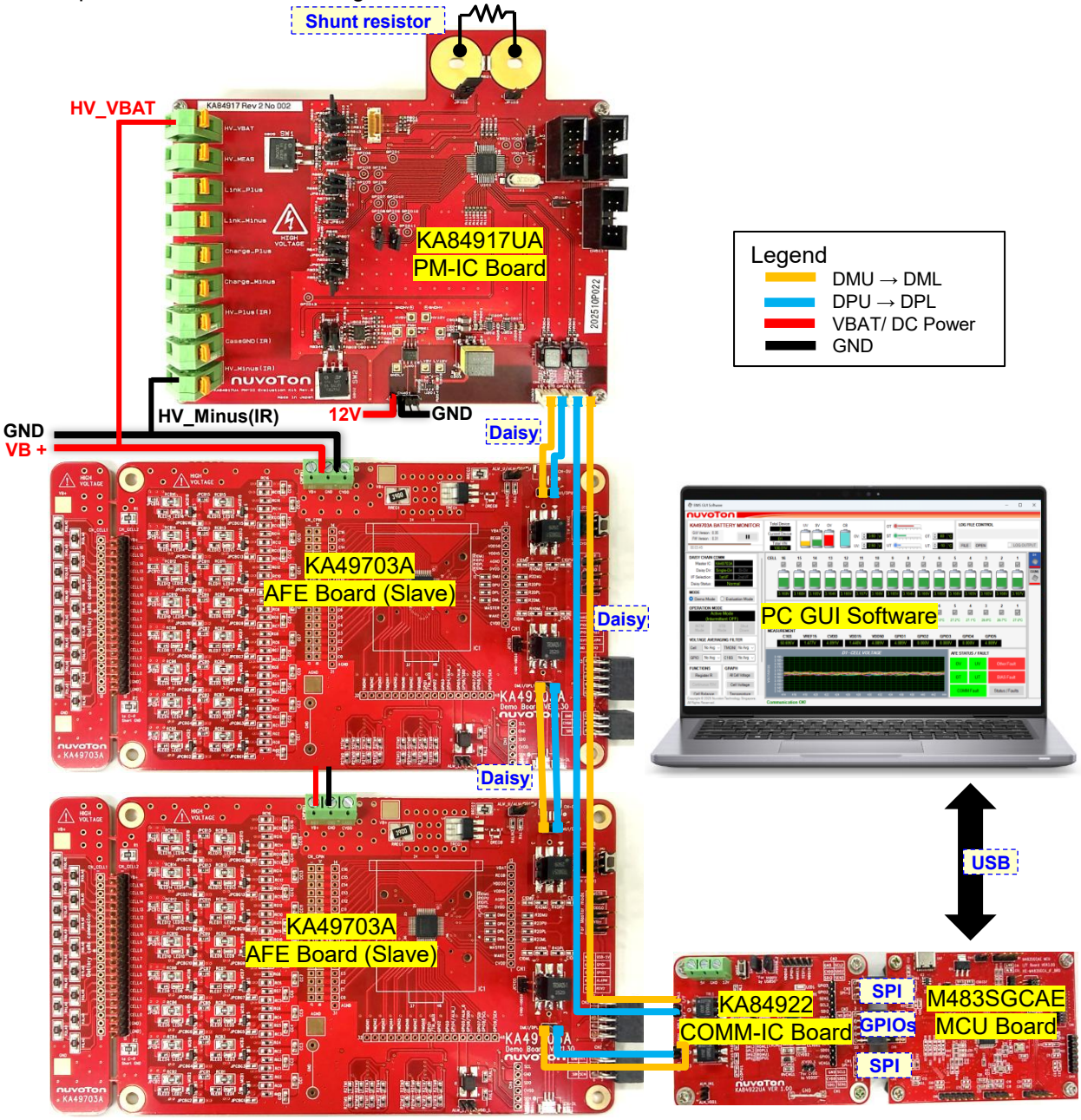


Figure 2.8 Setup of BMIC and PM-IC Configuration with COMM-IC

Daisy Connection	1st Interface: DMU (COMM-IC Master) ⇔ DML (AFE Slave) DPU (COMM-IC Master) ⇔ DPL (AFE Slave) DMU (AFE Slave) ⇔ DML (CN302 of PM-IC board) DMU (AFE Slave) ⇔ DPL (CN302 of PM-IC board)
	2nd Interface: DMU (CN301 of PM-IC board) ⇔ DML (COMM-IC Master) (loop back) DPU (CN301 of PM-IC board) ⇔ DPL (COMM-IC Master)

Table 2.3 Detailed daisy connections for BMIC and PM-IC configurations with COMM-IC

2.3 Hardware Connection for PM-IC Board Operation

2.3.1 Connection for Current Measurement

To perform current measurement using the PM-IC, two connection options are available:

i. **On-board Shunt Resistor Installation**
Mount a shunt resistor on R621 of PM-IC board to enable real current sensing. The PM-IC measures the voltage drop across the shunt resistor to calculate the current.

ii. **External DC Power Supply Simulation**
Connect a DC power supply to the shunt input terminals to simulate the voltage drop across a shunt resistor. The corresponding shunt resistor value can then be configured in the PC GUI to emulate actual current measurement behavior.

For detailed hardware connection instructions, refer to Chapter 2.2.2 – BMIC with PM-IC Configuration.

2.3.2 Connections for Voltage Measurement

To support various voltage monitoring and diagnostic functions, dedicated high-voltage (HV) measurement ports are provided on the board:

HV_VBAT – Used for battery pack total voltage measurement.

HV_MEAS / Link_Plus / Link_Minus / Charge_Plus / Charge_Minus – Used for voltage measurement of relay diagnosis function. These ports allow the system to monitor voltage differences across relay terminals to determine relay status (open/close) and detect potential faults.

HV_Plus (IR) – Used for voltage measurement of insulation resistance (IR) detection function.

The daisy chain system supports high-voltage measurement applications. The board is designed to handle battery pack voltages up to 1200 V.

Three selectable operating modes are provided:

i) 48 V Mode – supports voltage measurement up to 72 V

ii) 400 V Mode – supports voltage measurement up to 600 V

iii) 800 V Mode – supports voltage measurement up to 1200 V

The appropriate jumper configuration must be selected according to the intended operating mode. Refer to Chapter 2.2.2 – BMIC with PM-IC Configuration for detailed jumper settings and connection instructions.

2.3.3 Connection for Insulation Detection function

To perform insulation resistance detection, connect an external battery junction box with insulation resistors (see figure below).

For insulation detection measurement:
- Connect **HV_Minus (IR)** to the battery negative terminal (low potential side).
- Connect **HV_VBAT** to the battery positive terminal (high potential side).
- Connect **HV_Plus (IR)** before the insulation resistor on the high-voltage side.
- Connect **CaseGND (IR)** to the ground node of the insulation resistor network.

To enable the insulation detection function:
- Short jumper **JP902** to pull up GPIO9 to control **SW2**

Refer to the example connection diagram below for a typical insulation detection configuration using PM-IC.

Insulation resistance measurement results are calculated and evaluated via the PC GUI.
*(Refer to **Chapter 3.5.3** for detailed information on insulation detection calculation and evaluation)*

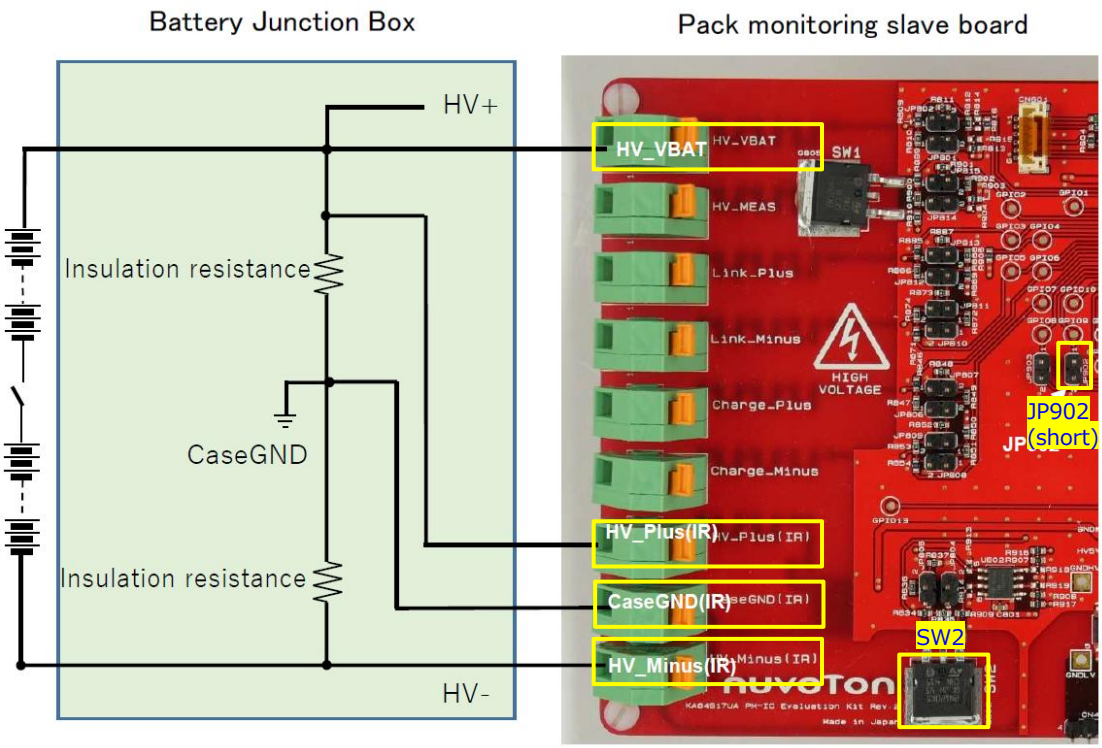


Figure 2.10 Example of connection for insulation detection function

2.3.4 Connection for VBAT/Relay Diagnostic Function

The PM-IC monitors the voltage at the high voltage terminals through resistor divider networks connected to ADC inputs. By comparing the voltage difference between these nodes, the system determines the actual relay state (Open or Closed). This function enables detection of relay faults, such as failure to open or close, thereby improving high-voltage system safety.

To connect the terminals as follows:
For total Pack Voltage (VBAT) Measurement
- Connect **HV_Minus (IR)** to the battery negative terminal.
- Connect **HV_VBAT** to the battery positive terminal.

For relay Status Diagnosis
- Connect **Link_Minus** and **Charge_Minus** to the corresponding low-potential measurement points.
- Connect **HV_MEAS**, **Link_Plus**, and **Charge_Plus** to the corresponding high-potential measurement points.

For detailed node locations (e.g., before/after fuse or relay), refer to the example connection diagram below.

To enable the relay diagnosis function:
- Short **JP903** to pull up GPIO8 to control **SW1**.

VBAT measurement and Relay diagnosis are performed through the PC GUI, where voltage values and relay status can be monitored and verified.
*(Refer to **Chapter 3.5.5** for detailed information on VBAT measurement and relay diagnosis)*

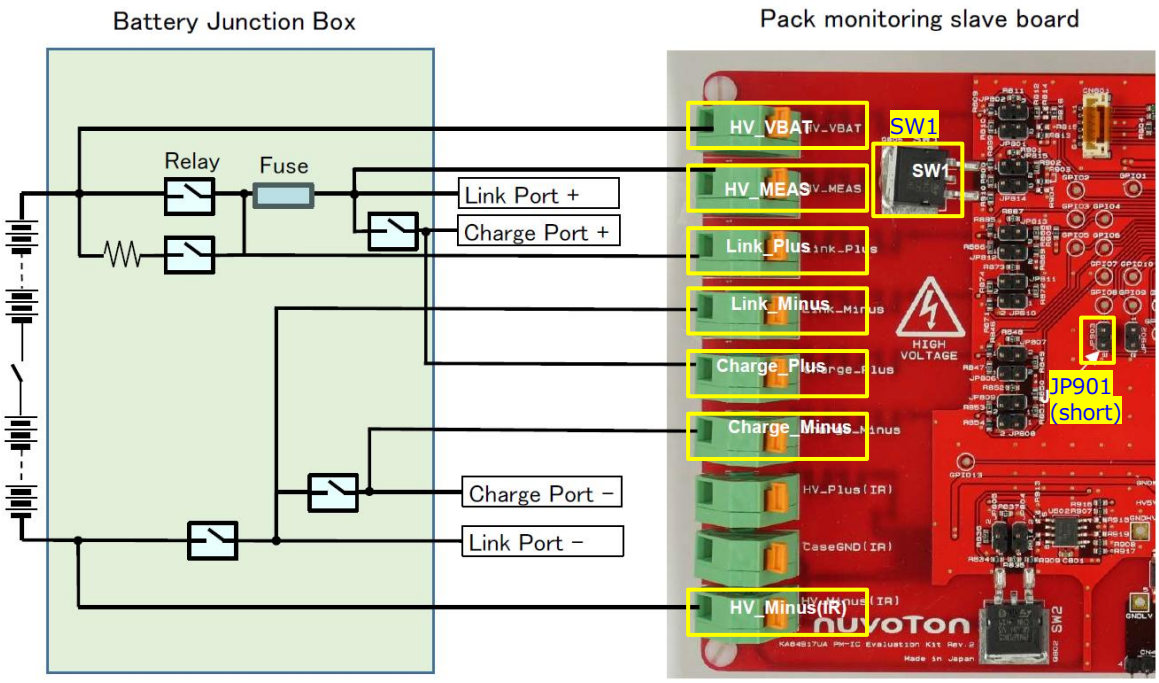


Figure 2.11 Example of connection for VBAT/Relay diagnosis function

2.4 Power Up the Boards

After connecting the hardware as shown on the previous pages, follow the steps below to power up the boards:

1) Connect the MCU board to the PC via USB.
   - LED5 (Red) will light up → MCU powered.
2) Turn ON the power supply (12.5 V – 76.8 V) for master and slave AFE boards.
   - LED1–LED4 blink → AFE boards configured and awake.
3) Turn ON the power supply (12V) for PM-IC board.
4) Open the application.
   - Wait for the USB COM port connection.
   - Click “Launch GUI” to enter the main window.

Notes:

- Check all hardware connections carefully before operation to avoid potential damage.
- A notification message will be displayed if the USB COM port is disconnected.
- A warning dialog will appear if the MCU configuration does not match the actual hardware configuration.
- The default MCU configuration consists of two devices connected in a daisy chain with a single-direction first interface (master DMU/DPU ↔ slave DML/DPL) . The device count includes the PM-IC/COMM-IC when a PM-IC/COMM-IC board is connected.
- LED1–LED4 will blink only when the hardware configuration matches the default MCU configuration.
- If the hardware configuration is not compatible with the MCU configuration, LED1-LED4 will not blink at startup. The AFE boards can still be activated by reconfiguring the settings via the GUI.

Example:

When using single-direction second interface (master DML/DPL ↔ slave DMU/DPU), the hardware configuration differs from the default MCU settings. As a result, LED1–LED4 will not blink at startup. In this case, update the configuration in the GUI to activate the AFE boards.

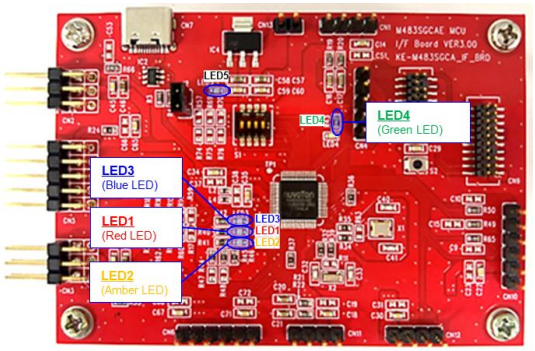


Figure 2.12 LED allocations and indications

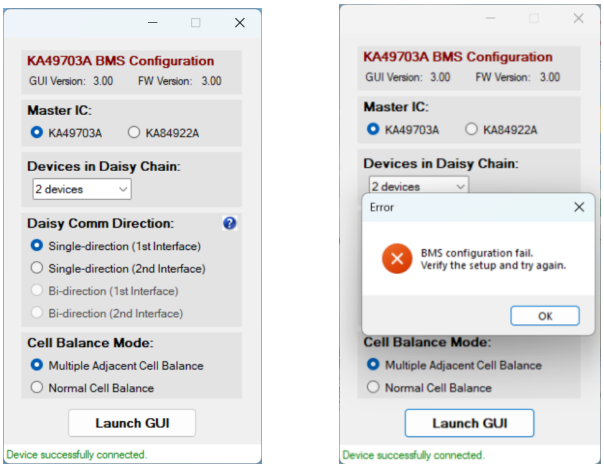


Figure 2.13 Configuration window
– connection success and failure examples

2.5 Firmware Update

This GUI includes a built-in firmware update feature, allowing users to easily update the firmware version when needed. Below are the steps to follow when a firmware update is required.

1) Click the latest version of the GUI application file.
2) A dialogue box will automatically appear to proceed with the firmware update if the flashed version of the MCU does not match.
3) The dialogue box will show progress while the firmware is being updated.
4) The firmware update takes only a few seconds to complete, after which the configuration window will display the updated GUI and firmware version numbers.

Note that a firmware update is required only when the flashed firmware is incompatible with the current GUI version. The entire update process takes less than a minute. The user only needs to wait until the “Launch GUI” button is enabled to start the main program.

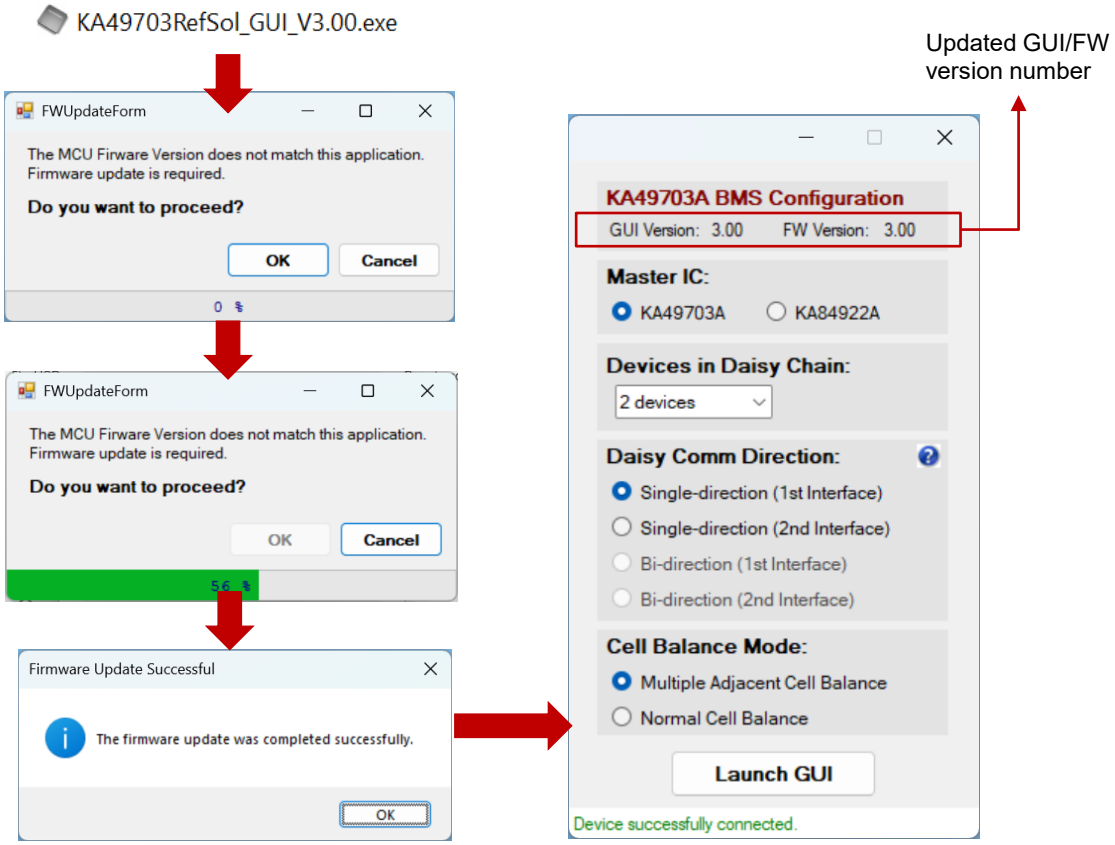


Figure 2.14 Firmware update procedure

3. PC GUI SOFTWARE DESCRIPTION

3.1 Configuration Window

The Configuration Window of the PC GUI software appears when the application starts, as shown in the figure below. It allows users to set up the initial BMS configuration settings before launching the main GUI window.

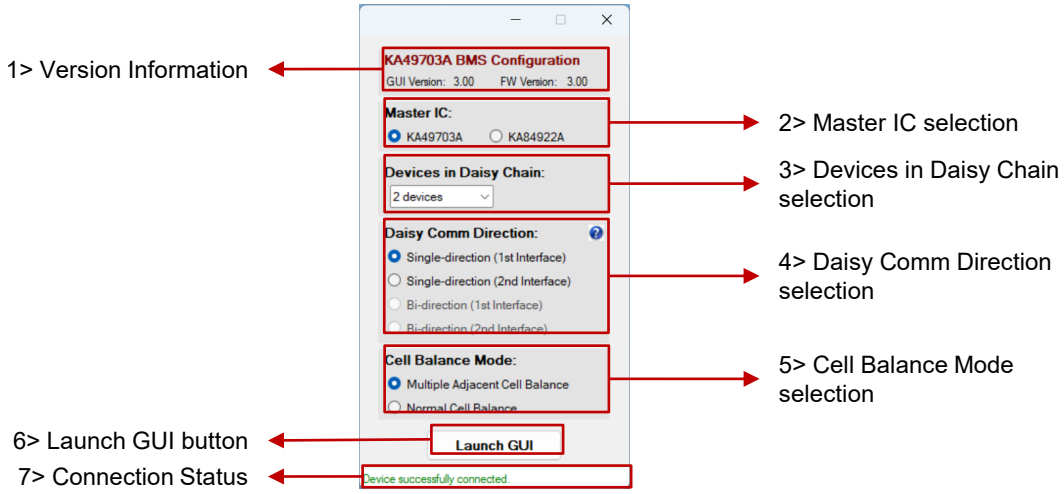


Figure 3.1 Configuration window

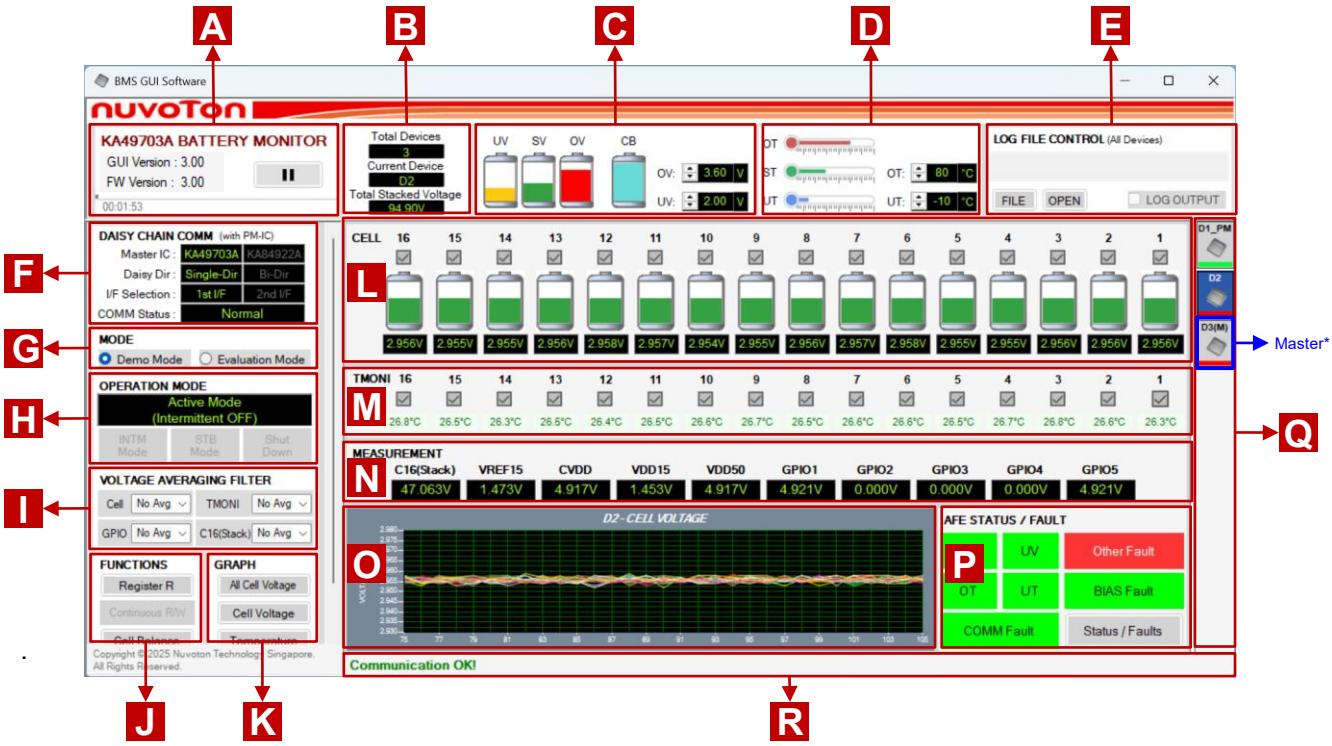
1) ***Version Information*** – displays the GUI and Firmware version numbers.
2) ***Master IC selection*** – selects master IC for current configuration.
3) ***Devices in Daisy Chain selection*** – defines the number of devices in the current daisy chain configuration (includes both master and slave devices). The GUI allows selecting 1 to 22 devices, with the default set to 2.
4) ***Daisy Comm Direction selection*** – configures the communication direction in the daisy chain.
   - Single-direction: Data flows only in one direction between devices.
     - 1st Interface: Master (DPU/DMU) → Slave (DPL/DML)
     - 2nd Interface: Master (DPL/DML) → Slave (DPU/DMU)
   - Bi-directional: Data can flow both ways between devices. The actual communication path depends on device address, interface configuration, and group communication settings.
     - 1st Interface: Used when the device address or configuration forces the upper path.
     - 2nd Interface: Used when the upper path is not forced (default to lower path).
5) ***Cell Balance Mode Selection*** – configures the cell balance according to the hardware setting.
6) ***Launch GUI button*** – opens the main GUI window after configuration is completed. Reconfigure the settings if necessary and try again.
7) ***Connection status*** – displays the USB COM port communication status. The Launch GUI button is enabled only when the USB COM port connection is successfully established.

3.2 Main Window

The Main Window of the PC GUI software, which appears when the application starts, is illustrated in Figure 3.2.1.

The PC GUI software communicates with the MCU via a USB COM port. It facilitates the transmission and reception of data to and from the BMIC through the MCU. Following data retrieval, the PC GUI software performs data conversion and presents the data in both graphical and text formats. Additionally, it serves as a reference guide for users, detailing the available system-level functions and operations.

This interface is segmented into several sections labelled A through R, each designated for specific functionalities. Further information regarding the purpose and operation of these sections will be elaborated on in the subsequent pages.



A - Application Info & Play/Pause Control
B - Device Summary
C - Cell Legend & OV/UV Setting
D - Temp legend & OT/UT Setting
E - Log File Control
F - Daisy Chain Communication
G - Mode
H - Operation Mode
I - Voltage Averaging Filter
J - Functions
K - Graph
L - Cell Voltage
M - TMONI
N - Measurement
O - Cell Voltage Graph
P - AFE Status / Fault
Q - Device Buttons
R - Communication Status

Master* : The label "Dx(M)" identifies the master device. When the master device is a COMM-IC, its corresponding device button is disabled.

Figure 3.2 Main window

Note: *The pause button is intended to pause the PC GUI data retrieval from the MCU, presenting a static data for convenient browsing. We do not guarantee the functionality of any operations during the pause state. When engaging in other activities, please click the "Pause/Play" button first to resume PC GUI data retrieval.*

A> Application Info & Play/Pause Control

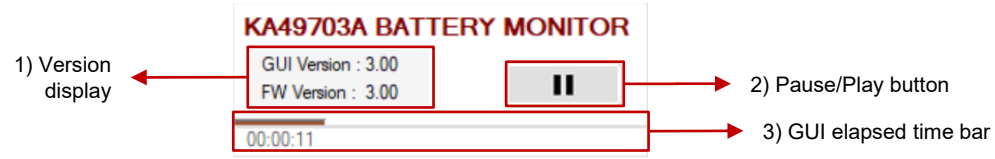


Figure 3.3 Main window - Application Info & Play/Pause Control

1) ***Version display*** – displays the PC GUI & MCU Firmware (FW) software version.
2) ***Pause/Play button*** – the button alternates between displaying "⏸" and "▶".
   - Click "⏸" to pause data acquisition of PC GUI from BMIC via MCU.
   - Click "▶" to play or resume data acquisition of PC GUI from BMIC via MCU when stopped.

   When clicking "⏸", some controls on the PC GUI will become greyed out, disabled or invalid. To click "▶" to re-enable these controls when needed.
3) ***GUI elapsed time bar*** – A progress bar that corresponds to the elapsed time when the PC GUI is opening and show the time in seconds.

B> Device Summary

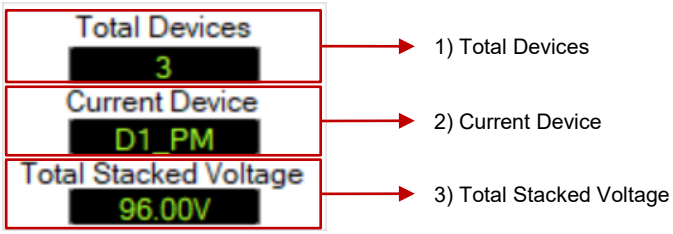


Figure 3.4 Main window – Device Summary

1) ***Total Devices*** – displays the total number of daisy-chained devices.
2) ***Current Device*** – displays the device currently selected for viewing. The cell voltages & graph, temperatures, measurement data, and fault status summary shown on the main window correspond to this device.
3) ***Total Stacked Voltage*** – displays the total voltage obtained by summing all battery cell voltages across the devices in the daisy-chain configuration.

C> Cell Legend & OV/UV Control

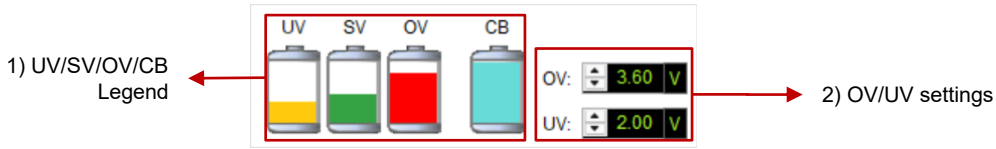


Figure 3.5 Main window - Cell Legend & OV/UV Control

1) ***UV/SV/OV/CB legend***– indicates the cell color image for under-voltage (UV) (orange), standard-voltage (SV) (green) , over-voltage (OV) (red) and cell balance (CB) (light blue).
2) ***OV/UV settings*** – sets over-voltage (OV) & under-voltage (UV) value for all daisy-chained device.

D> Temp legend & OT/UT Setting

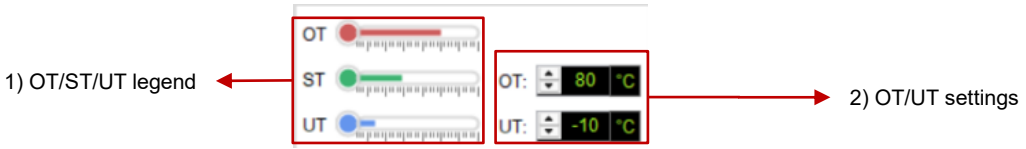


Figure 3.6 Main window - Temp legend & OT/UT Setting

1) ***OT/ST/UT legend*** – indicates the cell color image for over-temperature (OT) (red), standard-temperature (ST) (green) , under-temperature (UT) (blue).
2) ***OT/UT settings*** – sets over-temperature (OT) & under-temperature (UT) & value for all daisy-chained device.

E> Log File Control

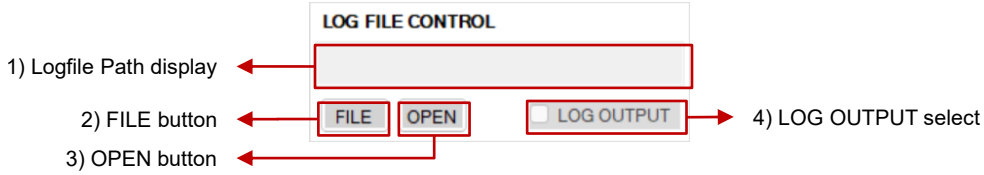


Figure 3.7 Main window - Log File Control

1) ***Logfile Path display*** – displays the log file path and file name.
2) ***FILE button*** – sets the log file location for storing data. By default, the files are saved in the 'LogFile' folder of the PC GUI software. Log files contain data only from the currently selected device. The file name indicates the device number associated with the logged data..

   The log file contents the converted & original HEX data retrieved from the BMIC through the MCU at regular intervals, every 1 second. The data includes cell voltage (CV1~CV16), temperature (TMONI1~16), C16S voltage, VREF15 voltage, VDD voltage, VDD15 voltage, VDD50 voltage, GPIO voltage (GPIO1~5). The following figure is an example of the data log output file, and the formatted default name of the file is “logFile-<Dx_Log_yyMMdd-HHmm>.csv”.

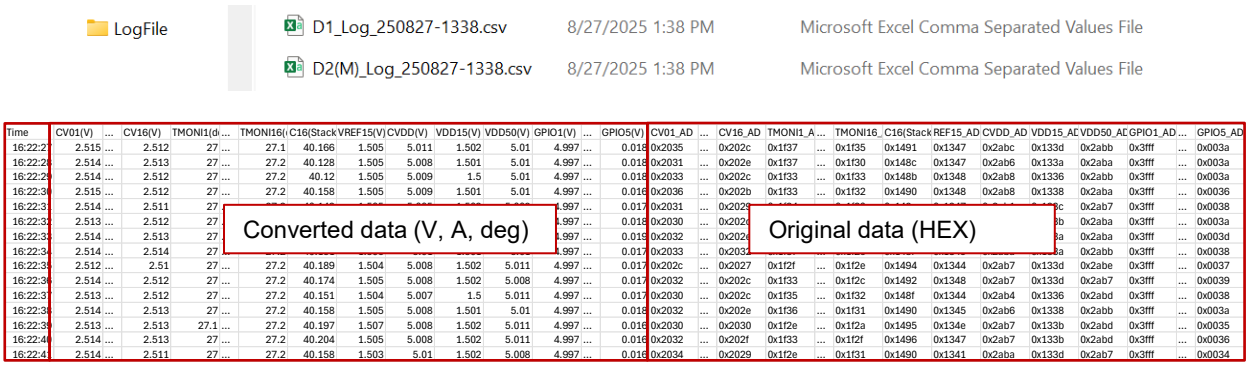


Figure 3.8 Example of the data log output file – All log data

3) ***OPEN button*** – opens the default location and select the log file (.csv). It's important to note that using this button to open the currently logged file is not recommended when the LOG OUTPUT is selected, as data cannot be logged while the log file is open.

4) ***LOG OUTPUT select*** – starts or stops the data log function, it can only be activated when a logfile location is chosen.

F> Daisy Chain Communication

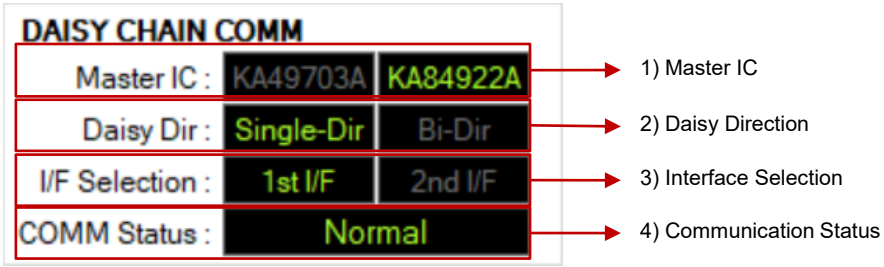


Figure 3.9 Main Window – Daisy Chain Communication

1) ***Master IC*** – displays the currently used master IC in the configuration.

2) ***Daisy Direction*** – displays the currently selected daisy-chain communication direction in the configuration. The highlighted option indicates selection, while the grayed-out option indicates unselected. For example, the above figure shows Single Direction selected, while Bi-Direction is grayed out.

3) **Interface Selection** – displays the currently selected 1st or 2nd interface in the configuration. The highlighted option indicates selection, while the grayed-out option indicates unselected. For example, the above figure shows the single direction 1st interface selected.

4) **Communication Status** – displays the status of daisy-chain communication.

- Normal (green) – communication is normal.
- Normal (Daisy Reconfig) (green) – communication is normal but has been reconfigured.
- AFE Comm Error (red) – communication error detected; a dialog will appear, and user action is required.

For example, in the figure below, when an AFE COMM Error is detected, the GUI continues to monitor the status for 2 seconds. If the error persists, a dialog box will be displayed. The user should check and fix the connection, then verify that the COMM status returns to Normal.

- If the status is normal, click OK to close the dialog box and continue.
- If the status is not resolved, click Cancel or Exit to restart the application.

Figure 3.10 Example of Daisy Communication Error happened

G> Mode

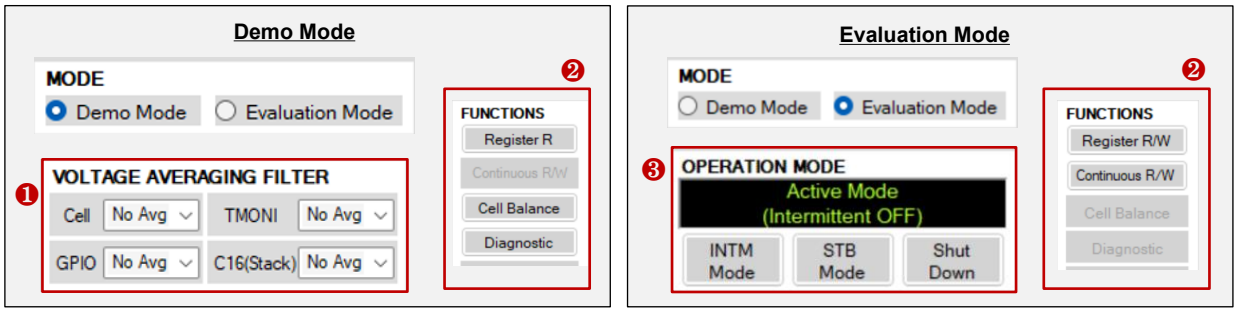


Figure 3.11 PC GUI appearance when Demo Mode & Evaluation Mode

Table 3.1.1 provides a summary of the PC GUI appearance changes when either demo mode or evaluation mode is selected.

	Demo Mode	Evaluation Mode
❶ Voltage Averaging Filter *	- Enabled - 4x averaging filters includes Cell / TMONI / GPIO / C16(Stack)	- Disabled
❷ Functions	- Some functions are enabled: • Register Read • Cell Balance • Diagnostic - Other functions are disabled (e.g Register Write / Continuous R/W)	- Some functions are enabled: • Register Read/Write • Continuous Read/Write - Other functions are disabled (e.g CB / Diagnostic)
❸ Operation Mode	- Disabled	- Enable - 3x buttons to control: Intermittent/Standby/Shutdown - 5x operation mode includes: Active No Intermittent / Active with Intermittent / Standby No Intermittent / Standby with Intermittent / Shutdown

Table 3.1 PC GUI appearance changes for Demo/Evaluation Mode

1) ***Demo Mode*** – To restrict users from directly modifying registers through the Register R/W sub window and Continuous R/W sub window, and to limit their access to operation.

2) ***Evaluation Mode*** – to open-up register write features, continuous read/write function, and operation mode. However, to mitigate the risk of potential malicious IC control, some demonstration functions such as Cell Balance, Diagnostic have been deactivated.

G> Operation Mode

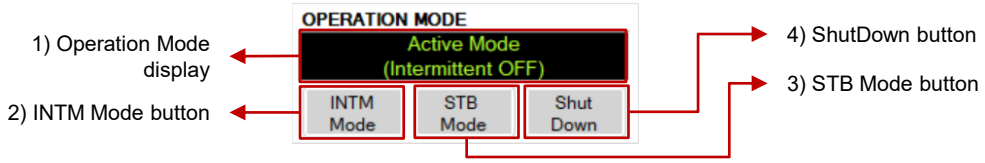


Figure 3.12 Main window – Operation Mode

1) ***Operation Mode display*** – to display the presently operation modes.

Operation Mode display	Register Control
"Active Mode (No Intermittent)"	MSET_SHDN 0x01[2] = "1" & MSET_STB 0x01[0] = "0"
	INTMSEL 0x01[11] = "0"
"Active Mode (Intermittent period xxxxxx msec)"	MSET_SHDN 0x01[2] = "1" & MSET_STB 0x01[0] = "0"
	INTMSEL 0x01[11] = "1" & INTM_TIM[2:0]
"Standby Mode (No Intermittent)"	MSET_SHDN 0x01[2] = "1" & MSET_STB 0x01[0] = "1"
	INTMSEL 0x01[11] = "0"
"Standby Mode (Intermittent xxxxxxxx sec)" "Shutdown"	MSET_SHDN 0x01[2] = "1" & MSET_STB 0x01[0] = "1"
	INTMSEL 0x01[11] = "1" & INTM_TIM2[2:0]
"Shutdown"	MSET_SHDN 0x01[2] = "0"

Table 3.2 Operation Mode display and Register Control

2) ***INTM Mode button*** – Click to turn the button ON and switch the BMIC to Active/Standby mode with Intermittent Mode(set INTMSEL 0x01[11] = "1"). Click again to turn the button OFF and return the BMIC to Active/Standby mode without Intermittent Mode.

3) ***STB Mode button*** – Click to turn the button ON and switch the BMIC to Standby mode (set MSET_STB 0x01[0] = "1"). Click again to turn the button OFF and return the BMIC to Active mode.

4) ***Shutdown button*** – Click to shut down the AFE (set MSET_SHDN 0x01[2] = "0"). This action shuts down the AFE board. Click again to turn the button OFF and return the AFE to Active mode.

The “Operation Mode” buttons are only enabled when in Evaluation Mode.

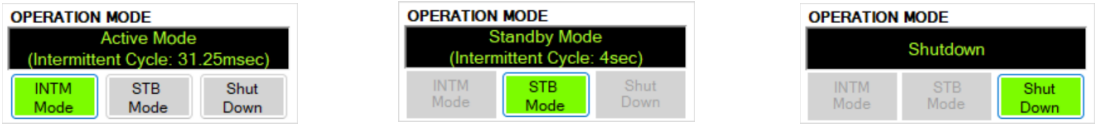


Figure 3.13 Examples of operation mode – Active Mode/Standby Mode/Shutdown

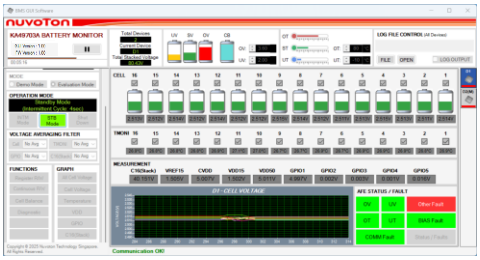


Figure 3.14 PC GUI appearance when Standy

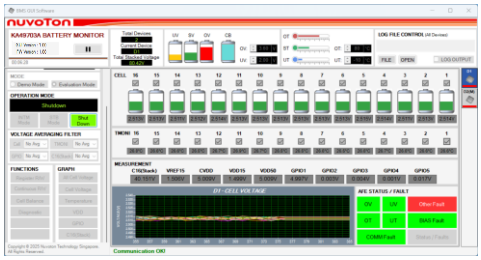


Figure 3.15 PC GUI appearance after Shutdown

I> Voltage Averaging Filter

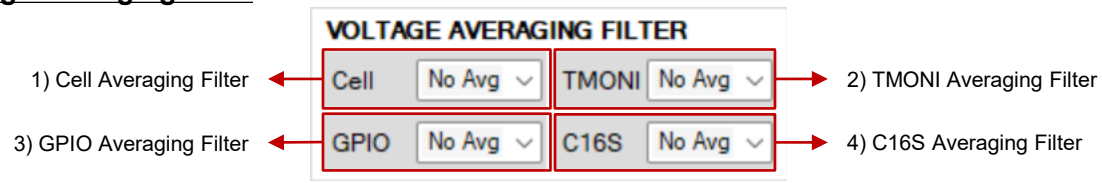


Figure 3.16 Main window – Voltage Averaging Filter

1) ***Cell Averaging Filter*** – sets ADV_AVE_CELL (address 0x08[15:14]) for number of averaging for cells voltage measurement.
2) ***TMONI Averaging Filter*** – sets ADV_AVE_TMONI (address 0x08[13:12]) for number of averaging for TMONI voltage measurement.
3) ***GPIO Averaging Filter*** – sets ADV_AVE_GPIO (address 0x08[11:10]) for number of averaging for GPIO voltage measurement.
4) ***C16S Averaging Filter*** – sets ADV_AVE_C16S (address 0x08[9:8]) for number of averaging for C16(Stack) voltage measurement

There are four options of voltage measurement average for all above settings:

- NO Avg – means no average, set data “00” (default)
- 4x Avg – means 4 times average, set data “01”
- 8x Avg – means 4 times average, set data “10”
- 16x Avg – means 4 times average, set data “11”

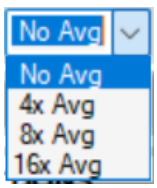


Figure 3.17 Drop down selection of Voltage Averaging Filter

J> Functions

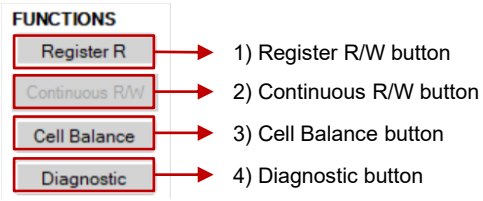


Figure 3.18 Main window – Functions

1) ***Register R/W button*** – opens the Register window for the current device to read or write BMIC registers, and view the register map of BMIC. The text on Register button displays “Register R” when in Demo Mode, but “Register R/W” when in Evaluation Mode.
2) ***Continuous R/W button*** – opens the Continuous Register Read/Write window for the current device to read or write BMIC registers continuously. The button is disabled when in Demo Mode.
2) ***Cell Balance button*** – opens the Cell Balance settings window for the current device. The button is disabled when in Evaluation Mode.
3) ***Diagnostic button*** – Opens the Diagnostic window for the current device. The button is disabled when in Evaluation Mode.

The details of sub windows are described in following Chapter3.3.

K> Graph

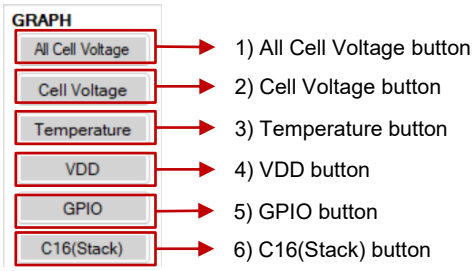


Figure 3.19 Main window – Graph

1) ***All Cell Voltage button*** – opens the graph window showing the cell voltages of all devices.
2) ***Cell Voltage button*** – opens the graph window showing the cell1~16 voltage of current device.
3) ***Temperature button*** – opens the graph window showing the TMONI1~16 value of current device.
4) ***VDD button*** – opens the graph window showing the VDD50, CVDD, VDD15, &VREF15 voltage of current device.
5) ***GPIO button*** – opens the graph window showing the GPIO1~5 voltage of current device.
6) ***C16(Stack) button*** – opens the graph window showing the C16(Stack) voltage of current device.

The details of sub windows are described in following Chapter3.3.

L> Cell Voltage

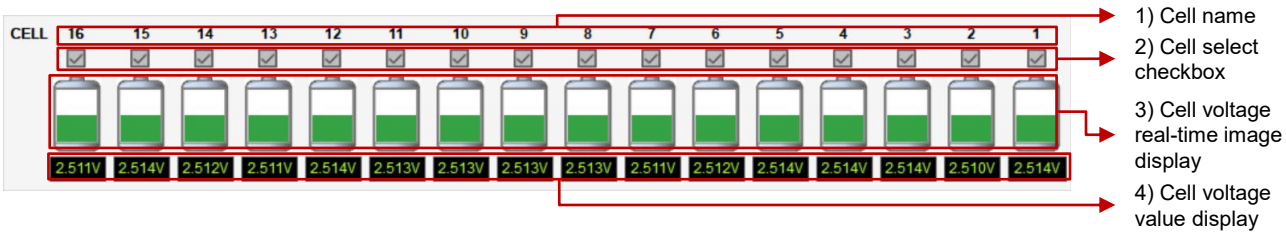


Figure 3.20 Main window – Cell Voltage

1) ***Cell name*** – Cell name of cell 1 to 16.

2) ***Cell select checkbox*** – switches ON/OFF cell voltage measurement for the current device individually:

   - Switch ON: checked (set corresponding bit of BMIC register 0x04 to “1”) .
   - Switch OFF: unchecked (set corresponding bit of BMIC register 0x04 to “0”).

3) ***Cell voltage real-time image display*** – displays timely visualization of dynamic cell voltage data retrieved from the BMIC via the MCU for the current device. Simultaneously, the statuses of UV, SV, OV and CB for the cells will be reflected.

4) ***Cell voltage value display*** – displays the timely cell voltage value for the current device in volts (V), which is retrieved and converted from BMIC via MCU.

   - The cell voltage value displayed is converted from readout data of KA49703A BMIC Register 0x2E ~ 0x3D.
   - The PC GUI will regularly retrieve data from the BMIC through the MCU, ensuring that all the information displayed above is kept up-to-date.

M> TMONI

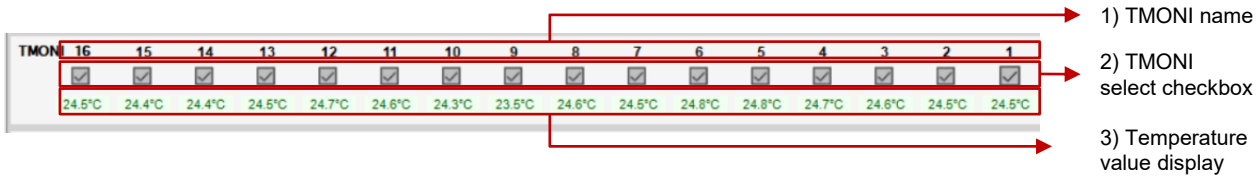


Figure 3.21 Main window – TMONI

1) ***TMONI name*** – TMONI name of cell 1 to 16.

2) ***TMONI select checkbox*** – switches ON/OFF TOMONI measurement for the current device individually:

   - Switch ON: checked (set corresponding bit of BMIC register 0x06 to “1”) .
   - Switch OFF: unchecked (set corresponding bit of BMIC register 0x06 to “0”).

3) **Temperature value display** - displays timely TMONI value for the current device in °C which is calculated and retrieved from MCU.

- The temperature value displayed is conversion data from readout data of KA49703A BMIC Register 0x40 ~ 0x4F.
- The PC GUI will regularly retrieve data from the BMIC through the MCU, ensuring that all the information displayed above is kept up-to-date.

The temperature conversion depends on VDD50 voltage, internal pull up resistor of TMONI pin, TMONI_AD voltage and thermistor characteristic.

N> Measurement

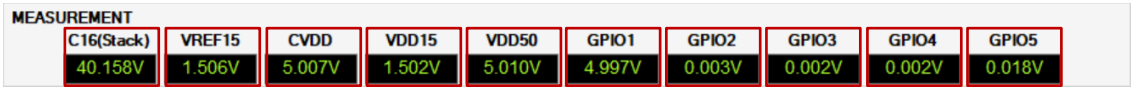


Figure 3.22 Main window – Measurement

1) ***C16(Stack) reading*** – Read data from BMIC Register 0x52 via MCU and convert it to voltage (V).
2) ***VREF15 reading*** – Read data from BMIC Register 0x58 via MCU and convert it to voltage (V).
3) ***CVDD reading*** – Read data from BMIC Register 0x59 via MCU and convert it to voltage (V).
4) ***VDD15 reading*** – Read data from BMIC Register 0x5A via MCU and convert it to voltage (V).
5) ***VDD50 reading*** – Read data from BMIC Register 0x5B via MCU and convert it to voltage (V).
6) ***GPIO1 reading*** – read data from BMIC Register 0x53 via MCU and convert it to voltage (V).
7) ***GPIO2 reading*** – Read data from BMIC Register 0x54 via MCU and convert it to voltage (V).
8) ***GPIO3 reading*** – Read data from BMIC Register 0x55 via MCU and convert it to voltage (V).
9) ***GPIO4 reading*** – Read data from BMIC Register 0x56 via MCU and convert it to voltage (V).
10) ***GPIO5 reading*** – Read data from BMIC Register 0x57 via MCU and convert it to voltage (V).

All the measurement data are read from the current device.

The voltage conversion is based on BMIC product standards.

The PC GUI will regularly retrieve data from the BMIC through the MCU, ensuring that all the information displayed above is kept up-to-date.

O> Cell Voltage Graph

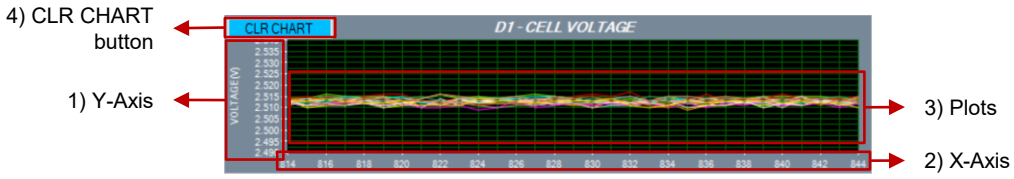


Figure 3.23 Main window – Cell Voltage Graph

1) ***Y-Axis*** – represents the cell voltage value in Volts (V). The Y-axis ranges from 0V to 5V.
2) ***X-Axis*** – represents the elapsed time with interval of 1 second between consecutive points. The maximum number of consecutive point is 4096.
3) ***Plot*** – plots the 16 sets of cell voltage of the current device data acquired from the BMIC via MCU at intervals of 1 second for visualization. The data sets used consist of the same cell voltage measurements mentioned earlier in section "C> Cell Voltage".
4) ***CLR CHART button*** – resets the chart by clearing the existing data sets and re-plotting them. This action is particularly helpful when the chart reaches its maximum capacity of consecutive points (4096).

This cell voltage graph is the same with the CELL graph mentioned in Chapter 3.2's sub-window, with the distinction that sub window offers more control over the graph.

P> AFE Status / Fault

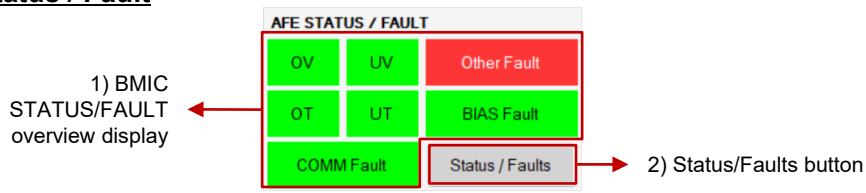


Figure 3.24 Main window – AFE Status / Fault

1) ***BMIC STATUS/FAULT overview display*** – to display the overview of status & faults for the current device. Read the register STAT2 (address 0x2D[6:0]).
   - Green – means all bits in the status/flag register are "0".
   - Red – means any bit in the status/flag register turns to "1".
2) ***Status/Faults button*** – opens the Status & Fault window for viewing the status of the BMIC status & fault of the current device.
   - Status page
   - Fault page

The status sub-window retrieves data from the BMIC status register via the MCU and presents information related to cell status, ongoing operations, and any detected faults.

The PC GUI will regularly retrieve data from the BMIC via the MCU, ensuring that all the information displayed above is kept up to date.

i. Fault page – To display the fault registers. The table below presents the corresponding register addresses for each group of functions.

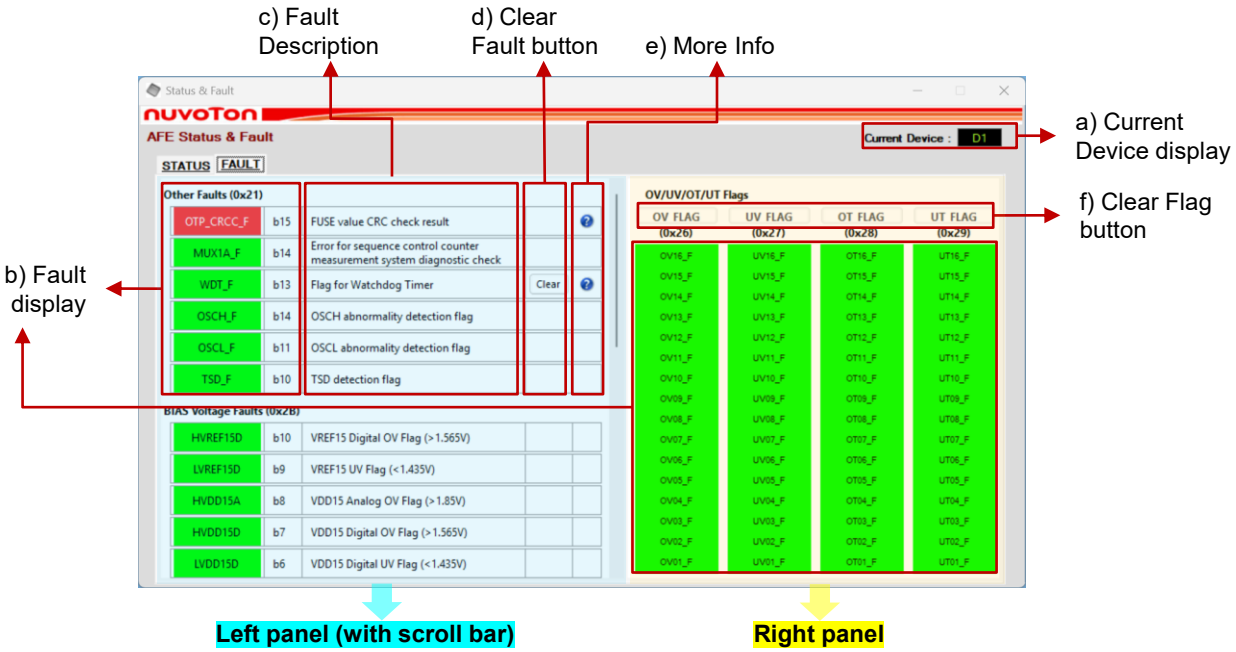


Figure 3.25 AFE Status / Fault – Fault page

Left panel (with scroll bar) – displays the status of the fault registers 0x21, 0x2B of the current device.

Right panel – display the status of fault registers 0x26, 0x27, 0x28, 0x29 of the current device.

a) Current Device display – shows the AFE status/fault of the current selected device.

b) Fault display – displays the status of the fault registers in the left and right panels above.

- Green label – means all bits in fault register are “0”.
- Red label – means any bit in fault register turns to “1”.

for example: OV16_F => register OV16_F 0x26[b15] is “0”

OTP_CRCC_F => register OTP_CRCC_F 0x21[b15] is “1”

c) Fault description – describes the meaning of the respective fault register.

d) Clear Fault button – clears the flag of the respective register when it turns to “1”.

e) More Info – clicks to show more explanation of the respective fault register.

f) Clear Flag button – clears all the flags of the respective register when any of the flags under this register turns to “1”. It includes OV_Flag, UV_Flag, OT_Flag, and UT_Flag buttons.

Note: The Clear Flag buttons are disabled by default. Their status is controlled by the Flag Latch Enable registers (0x0F[15:12]). To enable the Clear Flag buttons, set the Flag Latch Enable registers to “1”.

ii. Status page – To display the status of status register which are grouped by functions. The table below presents the corresponding register addresses for each group of functions.

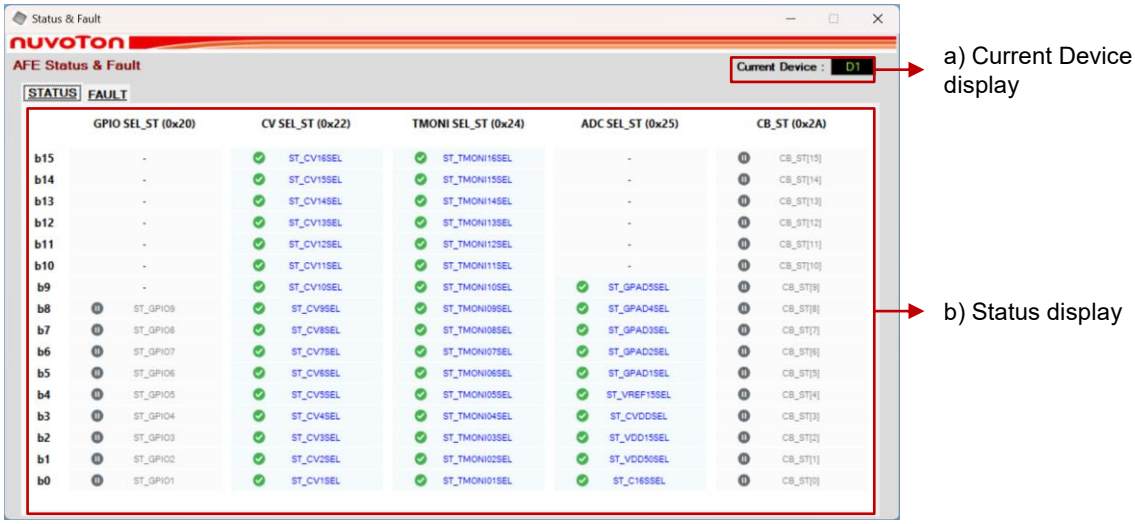


Figure 3.26 AFE Status / Fault – Status page

a) Current Device display – shows the AFE status/fault of current selected device.

b) Status display – displays status of the flag registers 0x20, 0x22, 0x24, 0x25, 0x2A of current device.

Grey label – means all bits in status/flag register are “0”.

Blue label – means any bit in status/flag register turns to “1”.

for example: ST_GPIO9 => register GPIO_SEL_ST 0x20[b8] is “0”

ST_CV16SEL => register CV_SEL_ST 0x22[15] is “1”

The PC GUI will regularly retrieve data from the BMIC through the MCU, ensuring that all the information displayed above is kept up-to-date.

Q> Device Buttons

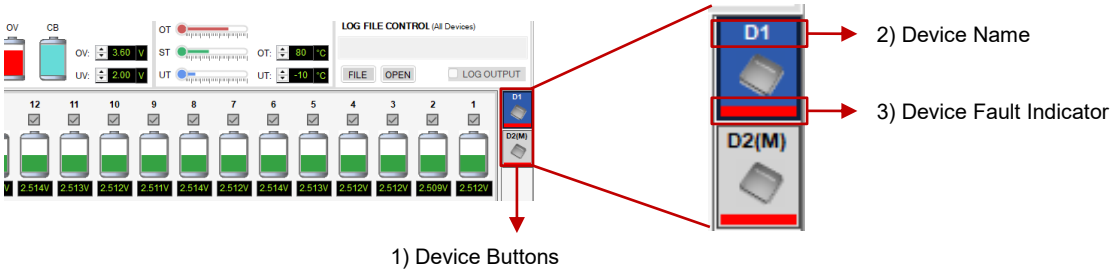


Figure 3.27 Main window – Device Buttons

1) ***Device buttons*** – click a button to display all measurement data from the corresponding device in the daisy-chain BMS system via the MCU, including raw register values and the converted or calculated parameters.

   The number of Device buttons depends on the number of devices configured initially. The maximum number of devices supported in this demo kit is 22. Each button represents one device in the daisy-chain BMS system.

2) ***Device Name*** – shows the address name of each corresponding device. Devices are listed from top to bottom in sequence. The label “Dx(M)” identifies the master device.

Note: If the master device is a COMM-IC, the corresponding device button is disabled.

2) ***Device Fault Indicator*** – Displays the fault status summarized from the STAT2 register (0x2D[6:0]) of the corresponding device.

- Green: No fault detected (all bits = “0”).
- Red: Fault detected (one or more bits = “1”).

R> Communication Status

Communication status shows whether the PC is successfully communicating with the MCU. If an error is detected, verify all connections and restart the application to re-establish communication.

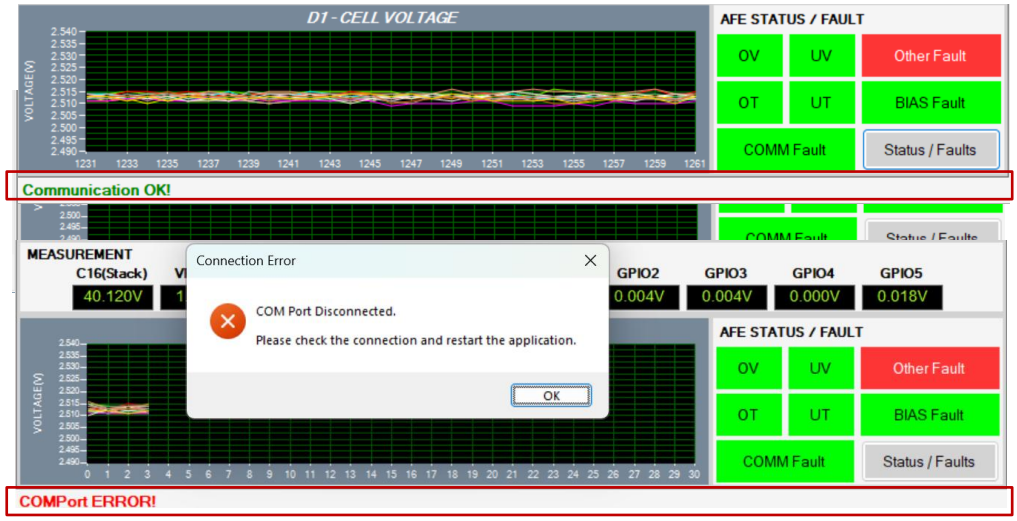


Figure 3.28 Main window – Communication Status

3.3 Sub Window

In addition to the main window, there are subsidiary windows integrated to facilitate specific functions.

The sub windows like Register Read/Write function, Continuous Read/Write function, Status display function, Cell balance setting function, Diagnostic check function and Graphs.

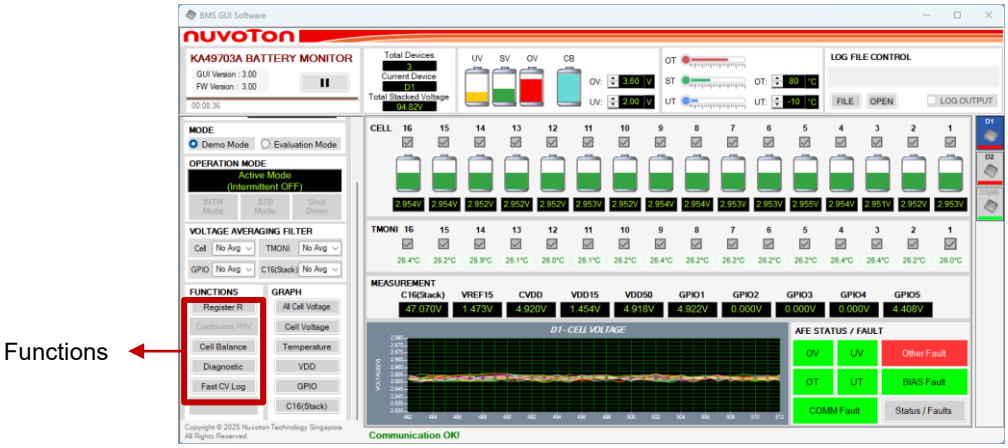


Figure 3.29 Functions buttons

3.3.1 Functions – Register R/W

The subwindow for Functions can be accessed from the main window using the following buttons.

The Register R/W sub window consists of 2 main parts:

- Register R/W – used to read data from or write data to BMIC registers via the MCU.
- Register Map – used to display the most up-to-date readout data for each bit for reference.

Refer to the details of each part on the following page.

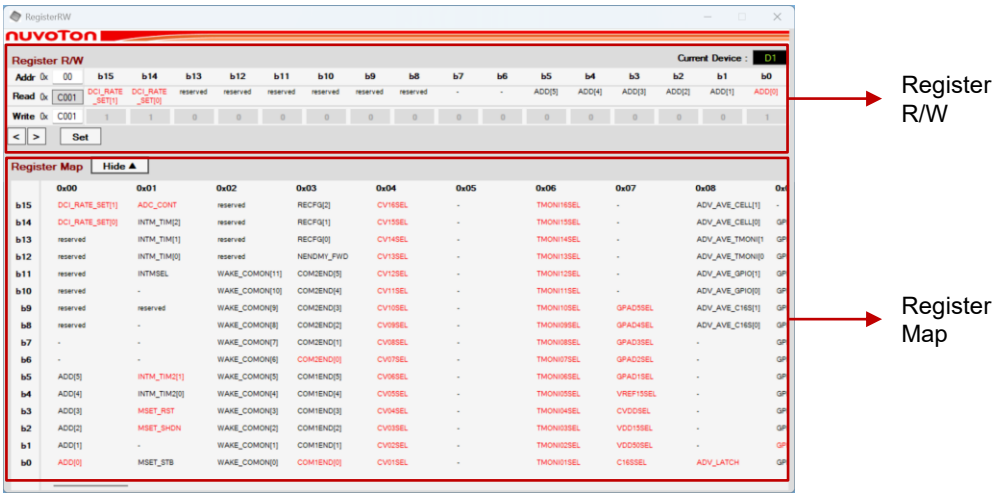


Figure 3.30 Sub Window – Register R/W Function

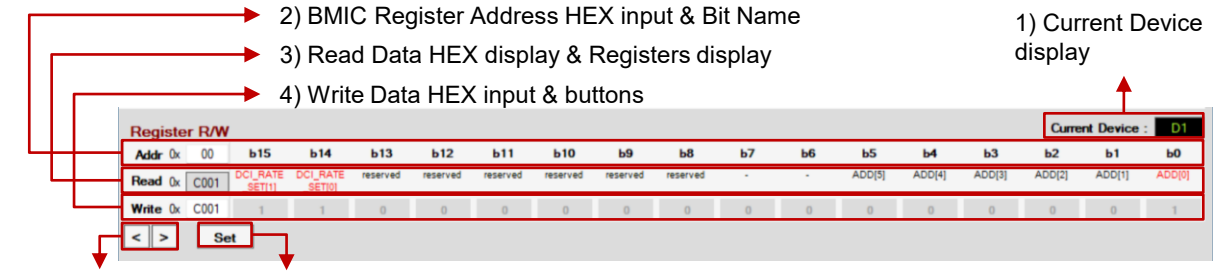


Figure 3.31 Register R/W – Register R/W control

1) ***Current Device display*** – shows the currently selected device; the register read/write operation applies to this device.

2) ***BMIC Register Address HEX input & Bit Name*** – to input a 1-byte hexadecimal number as the BMIC register address to read back BMIC data via the MCU. The bit names for the register address are designated as "b15~b0".

3) ***Read Data HEX display & Registers display*** – to display the readout data from BMIC via MCU. The readout data format is a 2-byte hexadecimal number. BMIC registers and their status under the specified Register Address, are shown accordingly. The color of the text corresponds to the high (H) or low (L) bit of a specific register (Black : “0” & Red : “1”).

4) ***Write Data HEX input & Buttons*** – to input a 2-byte hexadecimal number as the BMIC register address to read back BMIC data via the MCU. The Buttons are used to set individual bits in the BMIC registers to “0” or “1”. Disabled buttons indicate that specific registers can only be read. The control of Write Data HEX input and buttons is interdependent.

5) ***</> buttons*** – are used to navigate the previous/next BMIC register.

6) ***Set button*** – to write the current data into a specific BMIC register via the MCU.

When data acquisition of PC GUI from BMIC via MCU is paused (Pause/Play button - “▶”), clicking the Set button by the user will resume the data acquisition.

Please be aware that items 4) and 6) are visible and functional exclusively in Evaluation Mode, while they remain hidden and inactive in Demo Mode.

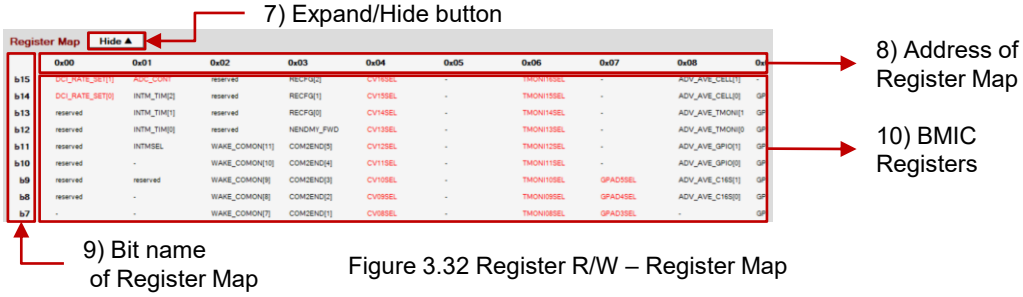


Figure 3.32 Register R/W – Register Map

7) ***Expand/Hide button -*** Click to expand/hide the register map display and register status.

8) ***Address of Register Map*** – Register address of KA49703A register map (0x01 to 0x5B).

9) ***Bit name of Register Map*** – Bit name of each register address (for 16 bits).

10) ***BMIC Registers*** – Registers of KA49703A BMIC. The color of the text corresponds to the high (H) or low (L) bit of a specific register (Black : “0” & Red : “1”).

The PC GUI will regularly retrieve data from the BMIC via the MCU, ensuring that all the information displayed above is kept up to date.

3.3.2 Functions – Continuous R/W

The Continuous R/W sub window is used to read/write data continuously to BMIC registers via MCU.

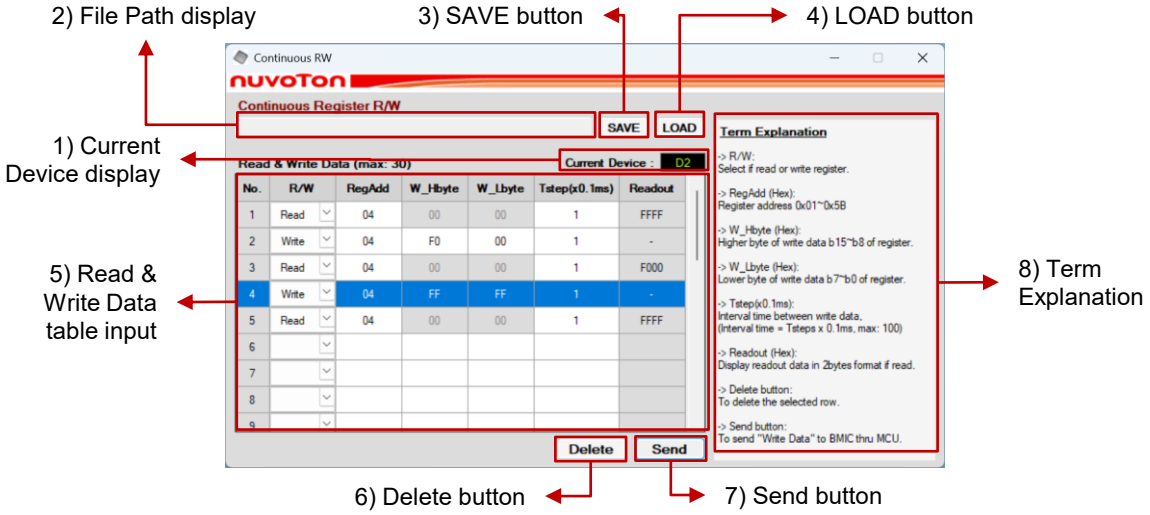


Figure 3.33 Sub Window – Continuous Register Read/Write Function

1) ***Current Device display*** – shows the currently selected device; the continuous register read/write operation applies to this device.

2) ***File Path display*** – to display the path/file name of the setting file.

3) ***SAVE button*** – to set the location to store the read/write data file and save the current read/write data into data file. The default location is in "ContRWData" folder within PC GUI software and the default name is "ContRW-<yyMMDD-HHmm>.txt".

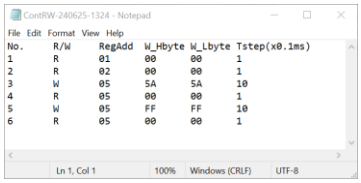


Figure 3.34 Example of read/write data file (.txt)

4) ***LOAD button*** – to open the read/write data file and load the read/write data.

The format of the text file should remain the same as the example above, adhering to the fixed formatting requirements.

5) ***Read & Write Data table input*** – to input sets of read/write data. Each set includes a data number, read or write selection, the register address, the higher byte and lower byte of write data, the interval time, and the readout data if read. The maximum sets of write data is 30.

6) ***Delete button*** – to delete the selected row in Read & Write Data table.

7) ***Send button*** – to send data of Read & Write Data table to BMIC via MCU.

8) ***Term Explanation*** – To explain the meaning of the terms that appear in the Read & Write Data table and provide guidance.

When data acquisition of PC GUI from BMIC via MCU is paused (Pause/Play button - "►"), clicking the Send button by the user will resume the data acquisition.

3.3.3 Functions – Cell Balance

The Cell Balance sub window is used to demonstrate the cell balance function of the BMIC. Users have the capability to configure Cell Balance ON/OFF settings with timing sequence control and send commands to the BMIC via MCU. This interaction results in the visual representation of cell performance in main window.

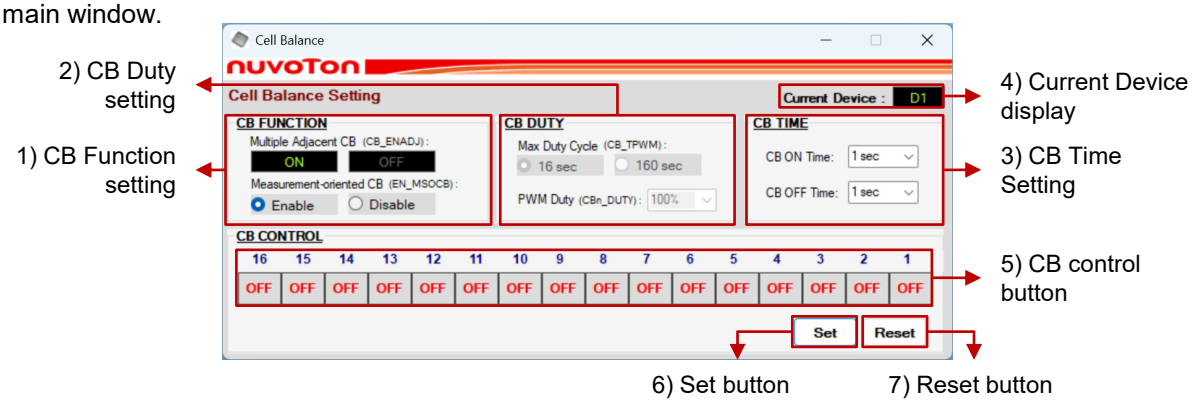


Figure 3.35 Sub Window – Cell Balance Function

1) ***CB Function setting*** – displays and sets the cell balance functions:

- Multiple Adjacent CB: ON/OFF indicates the current configuration.(This feature is configured in the Configuration window. Refer to Chapter 3.1 for details.)
- Measurement-oriented CB:
- enable means set EN_MSOCB (0x1A[6]) bit to “1”; CB Duty settings are disabled.
- disable means set EN_MSOCB (0x1A[6]) bit to “0”; CB Time settings are disabled.

2) ***CB Duty Settings*** – sets max duty cycle and PWM duty of cell balance settings.

- Max duty cycle: selects 16 secs, meaning set CBx_TPWM (0x1A[5]) bit to “0”; or selects 160 secs, meaning set bit to “1”.
- PWM duty: PWM duty level selection means set CBx_DUTY (0x1C ~ 0x1F) bits.

3) ***CB Time Settings*** – sets ON/OFF time for cell balance settings.

- CB ON Time: selects a CB ON duration from 1 to 15 secs in 1-sec steps.
- CB OFF Time: selects a CB OFF duration from 1 to 15 secs in 1-sec steps.

4) ***Current Device display*** – shows the currently selected device; the CB setting applies to this device.

5) ***Cell Balance Select buttons*** – consists of 16 buttons to set ON/OFF for individual channels during cell balancing. This setting corresponds to the CB_SEL register of BMIC.

- “OFF” means the specific channel is not selected for cell balancing.
- “ON” means the specific channel is selected for cell balancing, i.e., the corresponding CB_SEL[x] bit is set to “1”.

6) ***Cell Balance Reset button*** – the reset button is enabled only after cell balance is set. Click the button to reset the cell balance settings and send data to BMIC via MCU.

7) ***Cell Balance Set button*** – to send the cell balance settings to BMIC via MCU.

When data acquisition of PC GUI from BMIC via MCU is paused (Pause/Play button - “►”), clicking the Reset or Set button by the user will resume the data acquisition.

Cell Balance setting sequence

a. Enable/disable the measurement-oriented CB function.

b. Set the CB duty if the measurement-oriented CB function is disabled.

c. Set the CB time if the measurement-oriented CB function is enabled.

d. Click cell ON/OFF buttons (1~16) to select the CB channels.

e. Click “Set” button to send CB settings.

f. Click “Reset” button to reset CB settings.

The CB settings apply only to the current device. If the device number is changed by clicking the “device button”, the CB settings will be reset and the CB settings window will close.

The following examples show the results when the measurement-oriented CB setting is enabled or disabled, along with the corresponding performance.

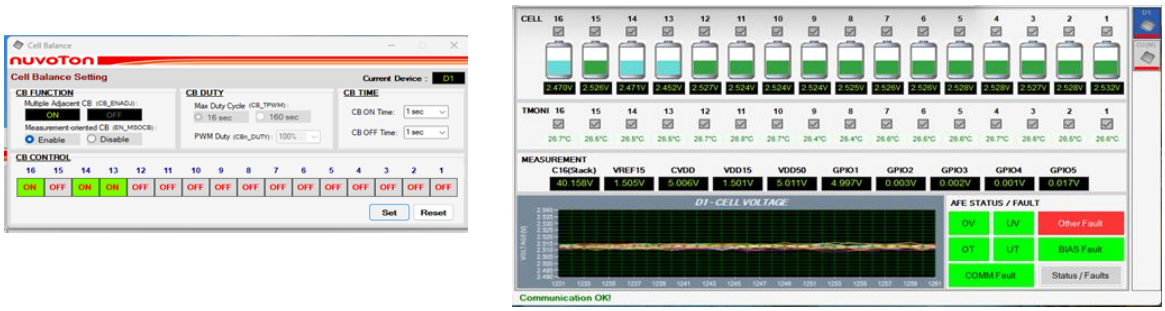


Figure 3.36 Example of visual representation of Cell Balance with Measurement Oriented CB function

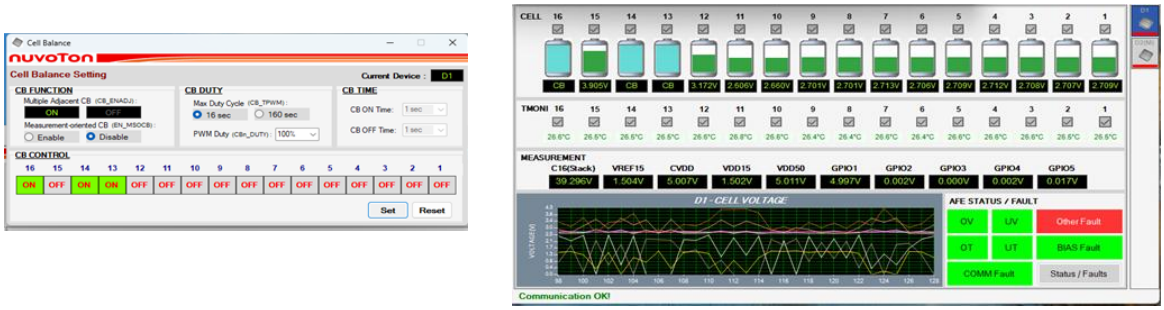


Figure 3.37 Example of visual representation of Cell Balance without Measurement Oriented CB function

Error Message

The PC GUI displays an error message if the user performs an incorrect action or if a fault occurs. Cell balancing cannot be performed while an error message is active. The following table lists the error messages and their possible causes, along with the recommended actions for the user to continue cell balancing.

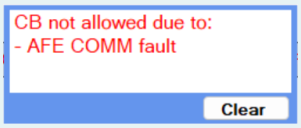
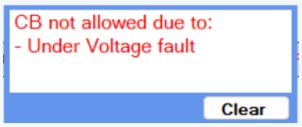
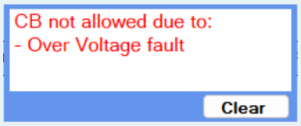
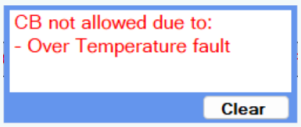
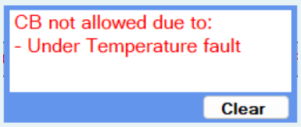
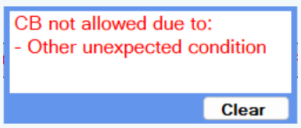
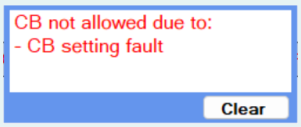
Pop-up alert message	Possible causes
	Communication error between AFE board and MCU board
	Under-voltage condition detected
	Over-voltage condition detected
	Over-temperature condition detected
	Under-temperature condition detected
	Unexpected condition detected (not listed above)
	Incorrect or invalid cell balance setting detected

Table 3.3 Pop-up error message of Cell Balance Function

3.3.4 Functions – Diagnostic

The Diagnostic sub-window is used to demonstrate the BMIC's safety diagnostic function. Users have the capability to do a diagnostic check on BMIC which is described in datasheet of KA49703A. The results of diagnostic check will be displayed after checking for a particular item.

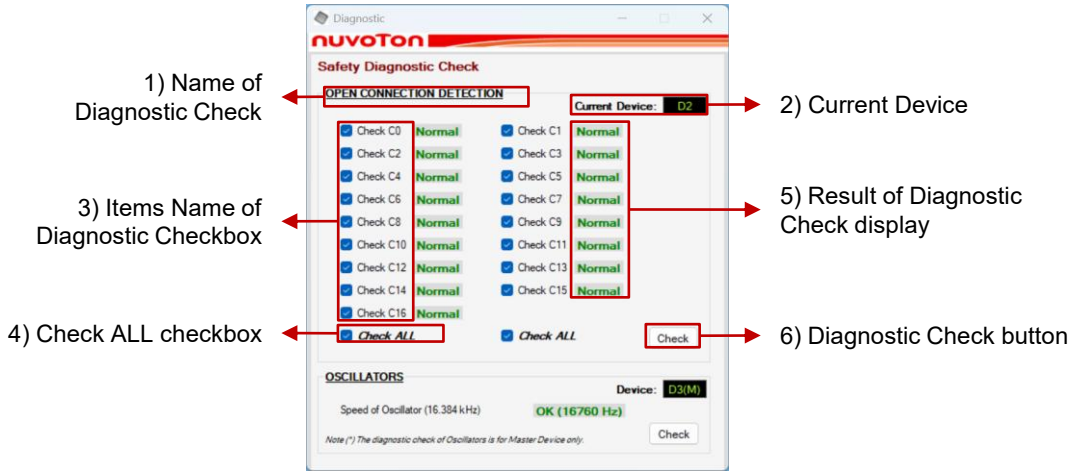


Figure 3.38 Sub Window – Diagnostic Function

1) ***Name of Diagnostic Check*** – specifies the name of the diagnostic check.

2) ***Current Device display*** – shows the device number under a specific diagnostic check. Noted that the diagnostic check of the oscillator is for the master device only.

3) ***Item Name of Diagnostic checkbox*** – specifies the item name of the diagnostic check and to selects the item for checking.

4) ***Check ALL checkbox*** – selects all items above the checkbox for the specific diagnostic check.

5) ***Result of Diagnostic Check display*** – displays the result of the diagnostic check for each item. The result is displayed using colored text.

- Green – means OK, no abnormality detected
- Red – means NG, abnormal detected

6) ***Diagnostic Check button*** – sends the command to the MCU for specific diagnostic checking. Only one type of diagnostic check is allowed at one time.

Diagnose Items	**Result - OK** (Green)	**Result - NG** (Red)
OPEN CONNECTION DETECTION		
Check C0~C16	**Normal**	**Opened**
OSCILLATORS		
Oscillator (16.384kHz)	**OK + Frequency** <Within range of Frequency (+/- 5%)>	**NG + Frequency** <Out of range of Frequency (+/- 5%)>

Table 3.3.2 The details of Diagnostic Check

3.3.5 Functions – Fast CV Log

The Fast CV Log sub-window is designed to log cell voltage (CV) data at high speed, with logging intervals in the range of several tens of milliseconds. The maximum logging duration is up to 30 minutes. Both raw CV data and processed CV data can be saved as separate CSV files for further analysis.

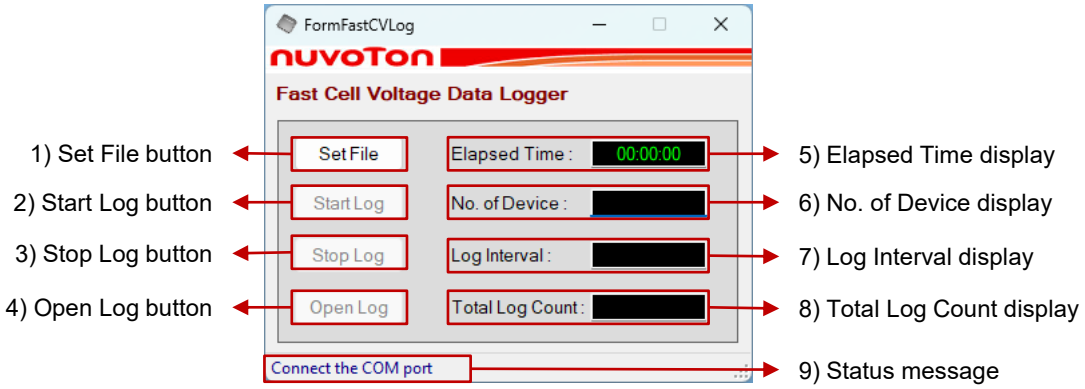
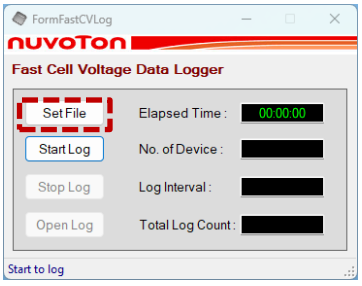


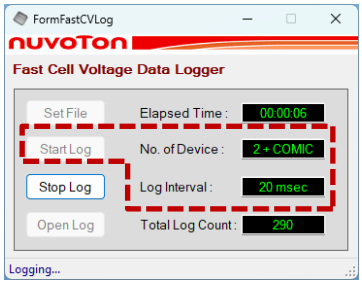
Figure 3.39 Sub Window – Fast CV Log Function

1) ***Set File button*** – allows the user to select and set the directory and file name for saving the logged data.
2) ***Start Log button*** – starts the CV data logging process.
   - The elapsed time counter begins running.
   - The number of connected devices and the log interval are read back from the MCU and displayed.
   - The total log count starts incrementing as data is logged.
3) ***Stop Log button*** – stops the CV data logging process.
   - The elapsed time counter and total log count are frozen.
   - After stopping, the user can open the log file to review the recorded data.
4) ***Open Log button*** – opens the log file from the specified directory for data inspection.
5) ***Elapsed Time display*** – shows the total logging time in seconds.
6) ***No. of Device display*** – displays the number of connected devices as read back from the MCU.
7) ***Log Interval display*** – displays the logging interval as determined by the firmware. The log interval depends on the number of connected devices:
   - ≤ 4 devices: approximately 20 ms
   - ≤ 8 devices: approximately 40 ms
   - ≤ 12 devices: approximately 60 ms
   - ≤ 16 devices: approximately 80 ms
   - ≤ 20 devices: approximately 100 ms
   - ≤ 22 devices: approximately 120 ms
8) ***Total Log Count display*** – shows the total number of data records logged during logging.
9) ***Status message*** – displays the current status of the CV data logging process (e.g., communication, logging, stopped, error).

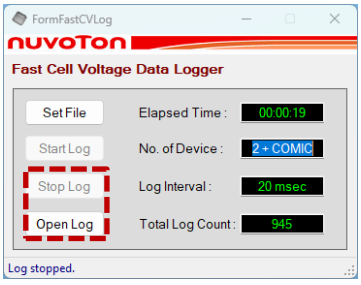
Fast CV data Logging steps:



i. Select file save location



ii. Start data logging
(No. of Device & Log Interval
will be readout and displayed)



iii. Stop data logging and open log file

Figure 3.40 Fast CV data Logging steps

Logged raw CV Hex data file and processed CV voltage data file:

LogCV_20260319_091840_raw.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	StartTime	9:18:50																			
2	DeviceCount	2																			
3	LogInterval	20 msec																			
4																					
5	IC Type	BMIC																BMIC			
6	DevAddr	Dev1																Dev2			
7	LogIndex	CV01_AD	CV02_AD	CV03_AD	CV04_AD	CV05_AD	CV06_AD	CV07_AD	CV08_AD	CV09_AD	CV10_AD	CV11_AD	CV12_AD	CV13_AD	CV14_AD	CV15_AD	CV16_AD	CV01_AD	CV02_AD	CV03_AD	CV04_AD
8	1	0x12AA	0x12A0	0x129E	0x12A6	0x12A6	0x12A2	0x12A4	0x12A8	0x12A2	0x12A7	0x12A8	0x12A4	0x12A9	0x12A3	0x12A5	0x12AA	0x12C5	0x12CA	0x12C5	0x12CD
9	2	0x12AA	0x12A0	0x129E	0x12A6	0x12A6	0x12A2	0x12A4	0x12A8	0x12A2	0x12A7	0x12A8	0x12A4	0x12A9	0x12A3	0x12A5	0x12AA	0x12C5	0x12C5	0x12C5	0x12CA
10	3	0x12A4	0x12A6	0x12A4	0x12A7	0x12A4	0x12A8	0x12A2	0x12A2	0x12AB	0x12A3	0x12A9	0x12A7	0x12A7	0x12A4	0x12A6	0x12AC	0x12CA	0x12C7	0x12C6	0x12C4
11	4	0x12A9	0x12A4	0x12AA	0x12A9	0x12A4	0x12A5	0x12A8	0x12A6	0x12A4	0x12A4	0x12A7	0x12A6	0x12A3	0x12A2	0x12A3	0x12AC	0x12C6	0x12C2	0x12C3	0x12C5
12	5	0x12A6	0x12A4	0x12A7	0x12A3	0x12A3	0x12A5	0x12A4	0x12A6	0x12A6	0x12A4	0x12A6	0x12A6	0x12A5	0x12AA	0x12A3	0x12A6	0x12C6	0x12C2	0x12C3	0x12C5
13	6	0x12A6	0x12A4	0x12A7	0x12A3	0x12A3	0x12A5	0x12A4	0x12A6	0x12A6	0x12A4	0x12A6	0x12A6	0x12A5	0x12AA	0x12A3	0x12A6	0x12C7	0x12C8	0x12C7	0x12C8

LogCV_20260319_091840_value.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	StartTime	9:18:50																			
2	DeviceCount	2																			
3	LogInterval	20 msec																			
4																					
5	IC Type	BMIC																BMIC			
6	DevAddr	Dev1																Dev2			
7	LogIndex	CV01(V)	CV02(V)	CV03(V)	CV04(V)	CV05(V)	CV06(V)	CV07(V)	CV08(V)	CV09(V)	CV10(V)	CV11(V)	CV12(V)	CV13(V)	CV14(V)	CV15(V)	CV16(V)	CV01(V)	CV02(V)	CV03(V)	CV04(V)
8	1	1.457	1.454	1.454	1.456	1.456	1.455	1.455	1.457	1.455	1.456	1.457	1.455	1.457	1.455	1.456	1.457	1.466	1.467	1.466	1.468
9	2	1.457	1.454	1.454	1.456	1.456	1.455	1.455	1.457	1.455	1.456	1.457	1.455	1.457	1.455	1.456	1.457	1.466	1.466	1.466	1.467
10	3	1.455	1.456	1.455	1.456	1.455	1.457	1.455	1.455	1.458	1.455	1.457	1.456	1.456	1.455	1.456	1.458	1.467	1.466	1.466	1.465
11	4	1.457	1.455	1.457	1.457	1.455	1.456	1.457	1.456	1.455	1.455	1.456	1.456	1.455	1.455	1.455	1.458	1.466	1.465	1.465	1.466
12	5	1.456	1.455	1.456	1.455	1.455	1.456	1.455	1.456	1.456	1.455	1.456	1.456	1.456	1.457	1.455	1.456	1.466	1.465	1.465	1.466
13	6	1.456	1.455	1.456	1.455	1.455	1.456	1.455	1.456	1.456	1.455	1.456	1.456	1.456	1.457	1.455	1.456	1.466	1.466	1.466	1.466

Figure 3.41 Example of logged raw CV Hex
data file and processed CV voltage data file

3.3.6 Graphs

The sub window for Graphs can be accessed from main window through the following buttons.

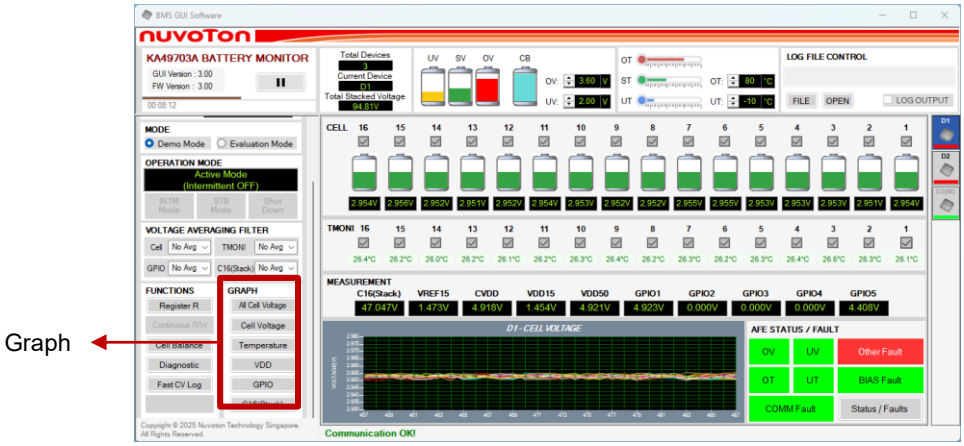


Figure 3.42 Graph buttons

All Cell Voltage Graph

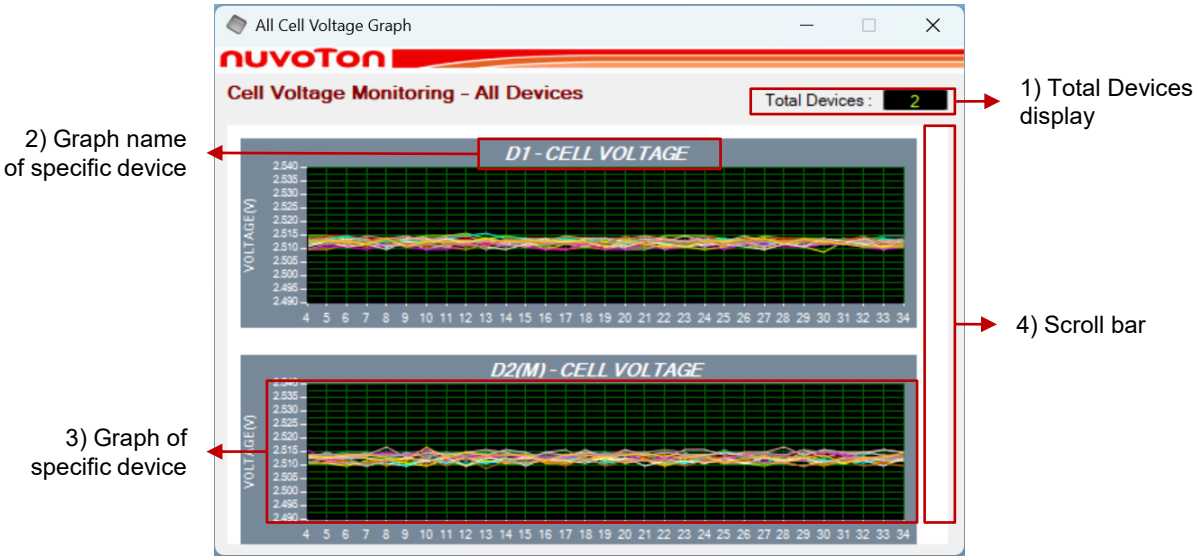


Figure 3.43 Sub Window – All Cell Voltage Graph

1) ***Total Devices display*** – displays the total number of devices, which determines the number of cell device graphs shown in this window.
2) ***Graph name of specific device*** – display the cell graph name of specific device.
3) ***Graph of specific device*** – display the cell graph of specific device.
4) ***Scroll bar*** – scrolls up and down to display all cell graph.

CELL VOLTAGE graph – to display the graph of Cells Voltage timely.

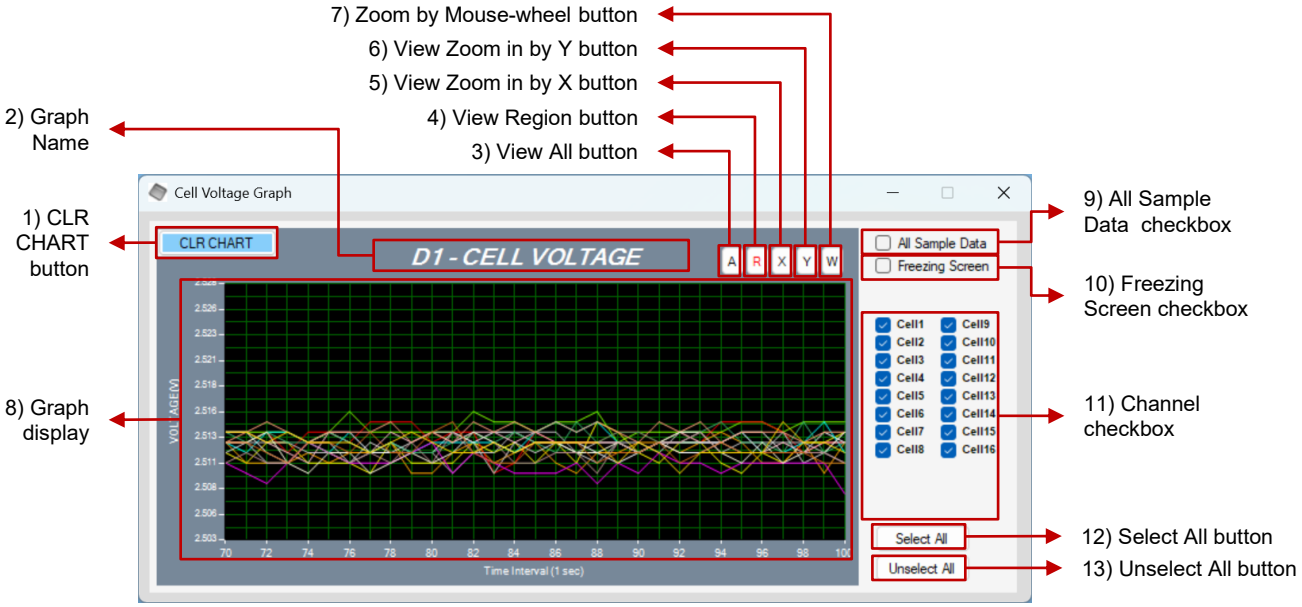


Figure 3.44 Sub Window – Cell Voltage Graph

1) ***CLR CHART button*** – clears the current chart.
2) ***Graph Name*** – displays the graph name of specific device.
3) ***View All button*** – views all data (Y axis).
4) ***View Region button*** – views by region.
5) ***View Zoom by X button*** – views zoom by X axis.
6) ***View Zoom by Y button*** – views zoom by Y axis.
7) ***View Zoom by Mouse-wheel button*** – zooms in/out by mouse wheel.
8) ***Graph display*** – displays the chart timely.
9) ***All Sample Data checkbox*** – selects whether the chart displays the most recent 30x sample data points or shows all sample data along the X-axis.
10) ***Freezing Screen checkbox*** – selects whether the chart displays real-time updates or remains static at a specific moment.
11) ***Channel checkbox*** – selects the displayed channels individually (16 channels).
12) ***Select All button*** – selects all channels.
13) ***Unselect All button*** – clears all selection of channels.

Note

The chart depicts data plotted over time, where each data point corresponds to the main window data updating interval of every 1s. The layout of all Graphs sub window are the same.

TEMPERATURE graph – to display the graph of temperature (TMONI1~16) timely.

VDD graph – to display the graph of VDD50, CVDD, VDD15 & VREF15 voltage timely.

GPIO graph– to display the graph of the GPIO voltage timely.

C16(Stack) graph – to display the graph of C16(Stack) voltage timely

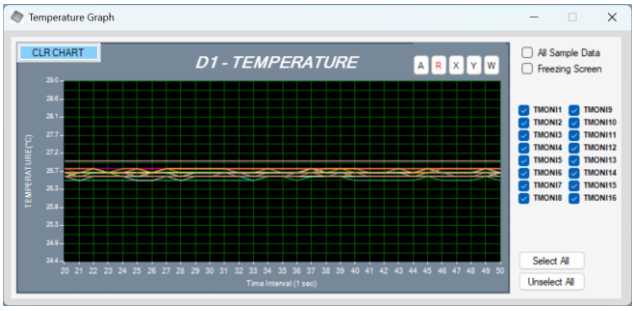


Figure 3.45 Sub Window – Temperature Graph

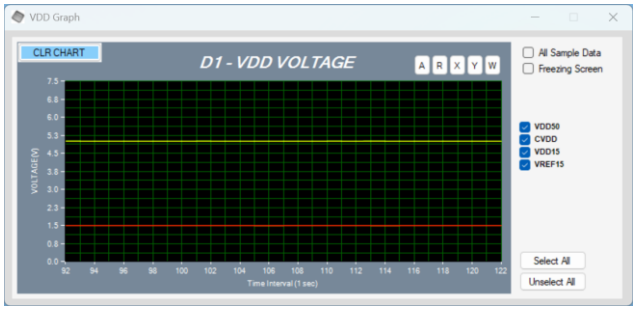


Figure 3.46 Sub Window – VDD Voltage Graph

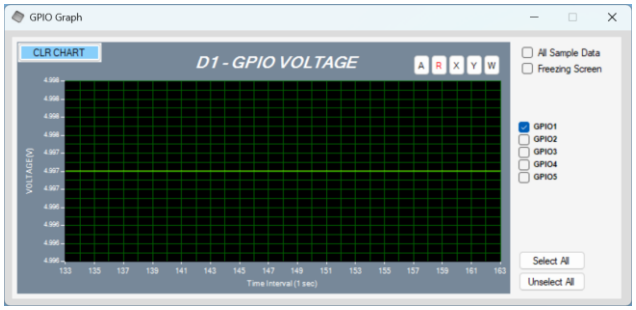


Figure 3.47 Sub Window – GPIO Graph

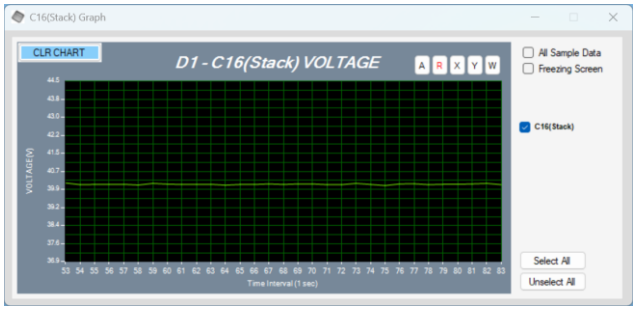


Figure 3.48 Sub Window – C16(Stack) Voltage Graph

Note

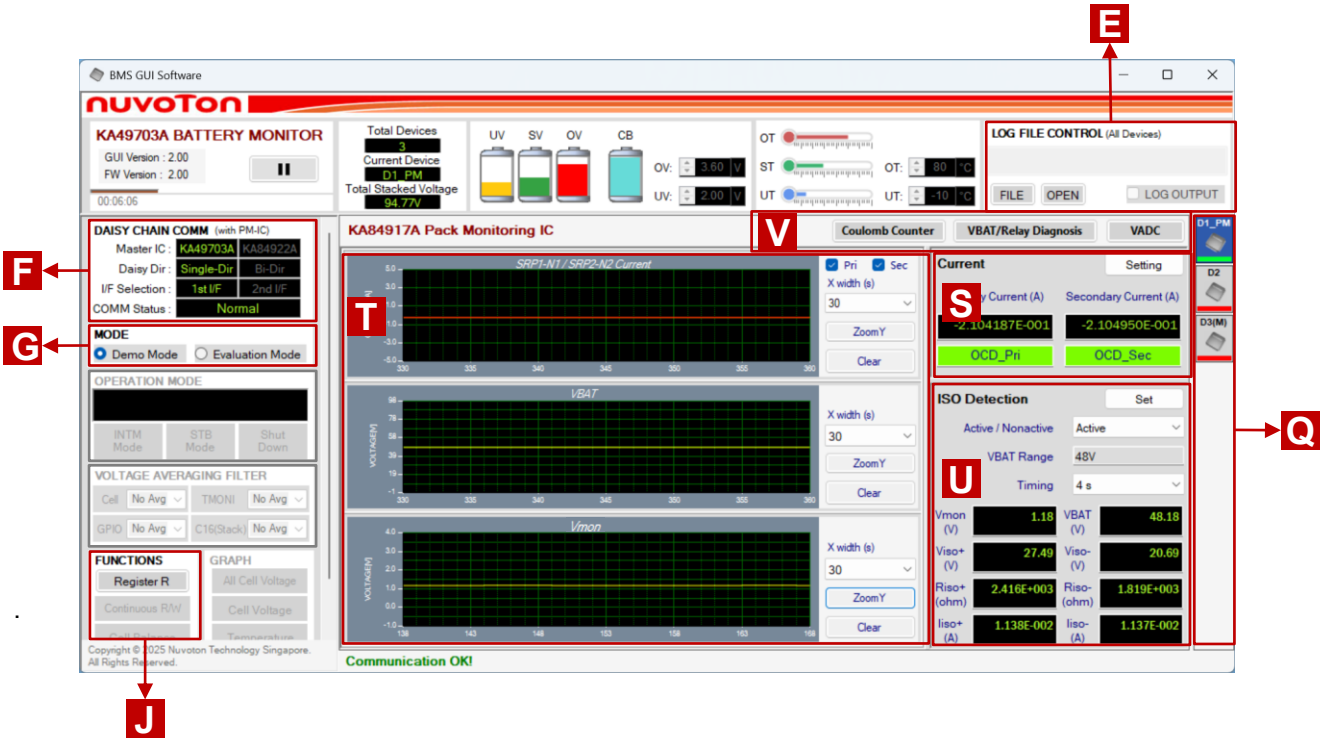
There are a total of sixteen channels (TMONI1~16) for BMIC temperature measurement. However, it's important to note that the number of temperature sensor (thermistor) used may vary depending on customization. In default configuration, our AFE board is equipped with 16 pcs of thermistors (NCU18XH103F60RB) for channel TMONI1~16. The temperature measurement within the PC GUI software is dependent on the characteristics of this standard thermistor component on our AFE board.

3.4 PM-IC Window

Similar to KA49703A, the PC GUI enables data transmission and reception between the PM-IC and MCU. Retrieved data is processed and displayed in both graphical and text formats. The GUI also supports coulomb counting, relay diagnosis, and insulation detection for system monitoring and evaluation.

The PM-IC Window of the PC GUI software, which appears upon selecting the PM-IC device, is illustrated in Figure 3.49.

The labelled segments below correspond to the PM-IC. Their purpose and operation are described in the following sections.



E – Log File Control (include PM-IC)
F – Daisy Chain Communication
G – Mode
J – Register R/W (PM-IC)
Q – Device Buttons
S – Current Measurement (PM-IC)
T – PM-IC Graph
U – Insulation Detection (PM-IC)
V – PM-IC functions

Note: Controls displayed in grey in the PC GUI are disabled or not applicable to the PM-IC.

Figure 3.49 Main window with PM-IC window

E> Log File Control (include PM-IC)

The PMIC log file records key parameters, including:

- Supply voltages (VDD50, VDDIO, VDD21, VDD18, VDD21MVSS31)
- Die temperature (DIETEMP)
- ADC measurements (VADC1–8, GPAD1–12)
- Battery voltages (VBATP, VBATN)
- Current and voltage FIFO data (FIFOOUT_C_P/S, FIFOOUT_V_P)
- Diagnostic-related data for VBAT/Relay diagnosis and insulation detection

An example of the data log output file is shown in the figure below.
The default file name format is:

logFile-<Dx_PM_Log_yyMMdd-HHmm>.csv

Refer to Chapter 3.3.1, Functions – Log File Control, for more details.

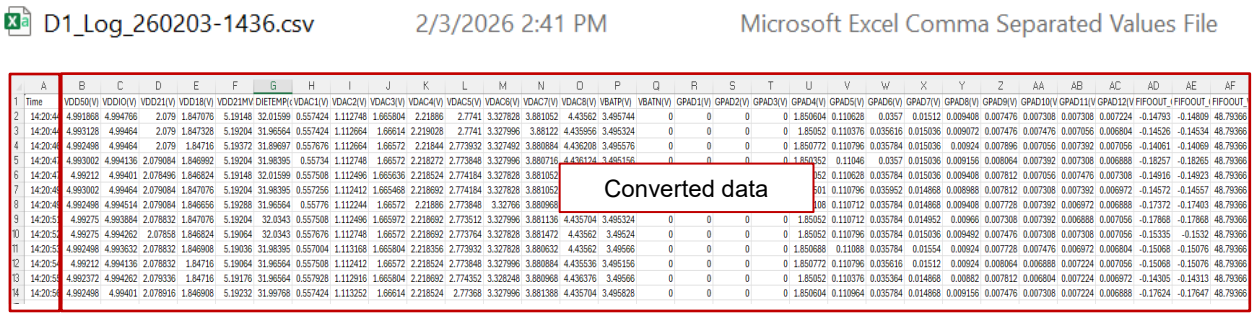


Figure 3.50 Example of the data log output file – PM-IC data

F> Daisy Chain Communication

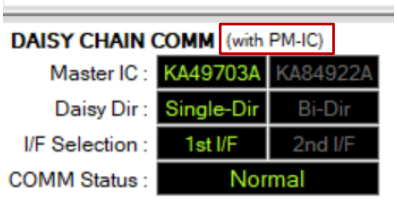


Figure 3.51 Main Window – Daisy Chain Communication with PM-IC

When the PM-IC KA84917A is connected, **"(with PM-IC)"** appears, while all other daisy chain communication functions remain unchanged. Refer to Chapter 3.3.1, Functions – Daisy Chain Communication, for more details.

G> Mode

Figure 3.52 Main Window – Daisy Chain Communication with PM-IC

When the PM-IC KA84917A is connected, users can access **Register Read** in Demo Mode and **Register Read/Write** in Evaluation Mode, in addition to the PM-IC functions.
Refer to Chapter 3.3.1, Functions – Mode, for more details.

J> Register R/W (PM-IC)

Register R/W function for the PM-IC KA84917A has the same operation as that for the BMIC KA49703A.
Refer to Chapter 3.3.1 Functions – Register R/W for more details.

Q> Device Buttons

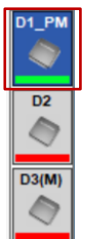


Figure 3.53 Main Window – Device Buttons with PM-IC

The device button for the PM-IC KA84917A is displayed as **Dx_PM**, where 'x' indicates the address number. The device fault indicator is not applicable to the PM-IC and will always appear in green. Refer to Chapter 3.2, Main Window – Device Buttons, for more details.

3.5 Sub Windows (PM-IC Functions)

3.5.1 Current Measurement

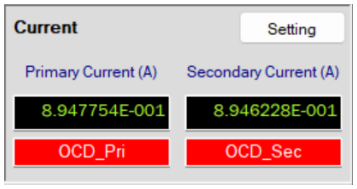


Figure 3.51 PM-IC Window – Current Measurement Not Configured

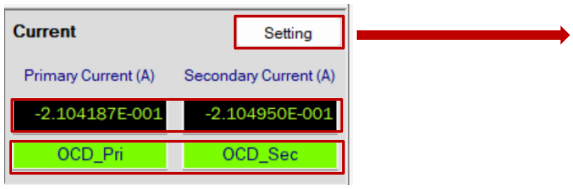


Figure 3.54 PM-IC Window – Current Measurement Configured

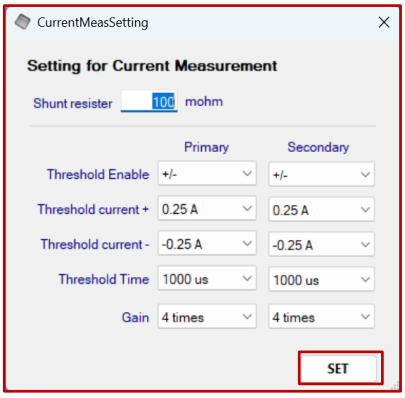


Figure 3.55 PM-IC Window – Setting window for current measurement

The current measurement function measures the battery pack's current.

After the current measurement parameters are properly set, the measured current values at the Primary and Secondary terminals are displayed in **amperes (A).** If the shunt resistor is not connected or no additional power supply is supplied, the current measurement setup is considered invalid, and any displayed information should be regarded as invalid.

Users can click the '**Setting**' button to enter the setting page, adjust the measurement parameters for current measurement purposes, and then press '**SET**' button to apply the settings.

The PM-IC KA84917A determines whether an overcurrent condition occurs based on the configured threshold values. The overcurrent detection status is indicated by the **OCD_Pri** and **OCD_Sec** LEDs:

- Green: Normal operation
- Red: Overcurrent detected (NG)

The setting page includes the following parameters for both Primary and Secondary terminals:

- Value of Shunt resistor on the PM-IC board(e.g: 100 mΩ)
- Threshold Enable (+/-, +, -, Disabled)
- Positive current threshold (0.25A ~ 1.025A, step size: 0.025A)
- Negative current threshold (-0.25A ~ -1.025A, steps size: -0.025A)
- Threshold time (100us ~ 1600us, steps size: 100us)
- Gain (4times, 16times, 64times, 256times)

Note: The positive and negative current threshold values change proportionally with the shunt resistor value. For example, when the shunt resistor is 1000 mΩ, the threshold range is +/-0.025A to +/-0.1025A.

3.5.2 PM-IC Graph

Figure 3.56 PM-IC Window – PM-IC Graphs

1) ***Current graph*** – Displays the primary (SRP1_N1) / secondary (SRP2_N2) current measurement value.

2) ***VBAT graph*** – Displays the VBAT value, calculated using the resistor divider in the PM-IC board circuit, along with the VBATP and VDD18 of the PM-IC register readout values.

3) ***Vmon graph*** – Displays the derived GPIO10_AD value used as the monitoring reading during insulation detection.

Note: The Y-axis of the three graphs represents the PC GUI monitoring time interval, which is 1 second.

4) ***Pri/Sec Current selection*** – Choose to display Primary Current, Secondary Current, or both.

5) ***X width setting*** – Sets the X-axis time span of the graph. (30s / 60s / 120s / 240s).

6) ***ZoomY button*** – Click to enable Y-axis zoom (button turns green), then drag the mouse on the graph to zoom in. Click again to exit (button returns to default color).

7) ***Clear button*** – Click to clear the graph and restart updating it.

Note: Features 5) – 7) are applied to each of the three graphs above.

3.5.3 Insulation Detection

This solution provides a demonstration of PM-IC-based insulation detection for battery pack using an exampled relay insulation circuit. Refer to Chapter 2.3.3 for the example circuit.

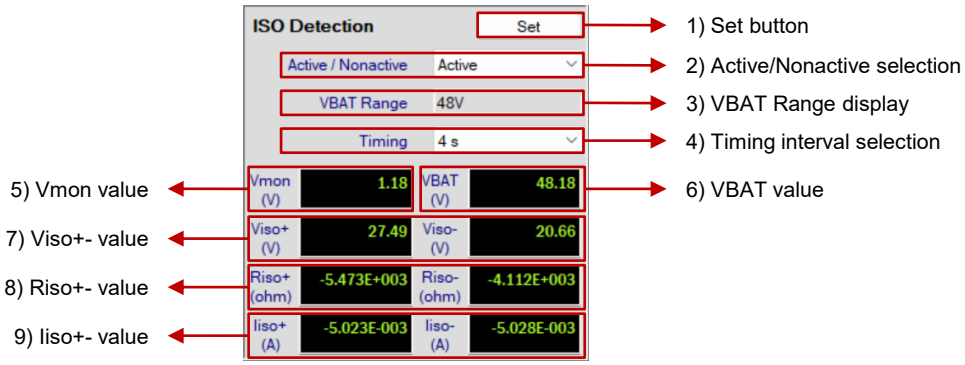


Figure 3.57 PM-IC Window – Insulation detection panel

1) ***Set button*** – Applies the settings and starts insulation detection after the PM-IC is set to Active mode. The insulation resistance value is then monitored.
2) ***Active/Nonactive selection*** – The Active option starts insulation detection, while the Nonactive option stops insulation detection.
3) ***VBAT Range display*** – Displays the VBAT range configured in the Insulation detection window.
4) ***Timing Interval Selection*** – Selects the timing interval for the insulation detection function, excluding Vmon. Available options are 4 s, 5 s, 6 s, and 7 s.
5) ***Vmon Value*** – Displays the GPIO10 readout value from the PM-IC, updated every second.
6) ***VBAT Value*** – Displays the VBAT value, which is the same as the VBAT graph value. This value is calculated using the resistor divider in the PM-IC board circuit, along with the VBATP and VDD18 register readout values from the PM-IC.
7) ***Viso± Value*** – Displays the calculated voltage across the insulation resistor.
8) ***Riso± Value*** – Displays the calculated resistance value of the insulation resistor.
9) ***Iiso± Value*** – Displays the calculated current flowing through the insulation resistor.

Note: The calculation details for Viso±, Riso±, and Iiso± are described in the following chapter.

The insulation resistance calculation is performed based on the circuits shown below, with GPIO13 controlling **SW3** on the PM-IC board to toggle ON and OFF, while **SW2** remains ON throughout the calculation. (SW3 and SW2 are switches of PM-IC board, refer to PM-IC schematic for details)

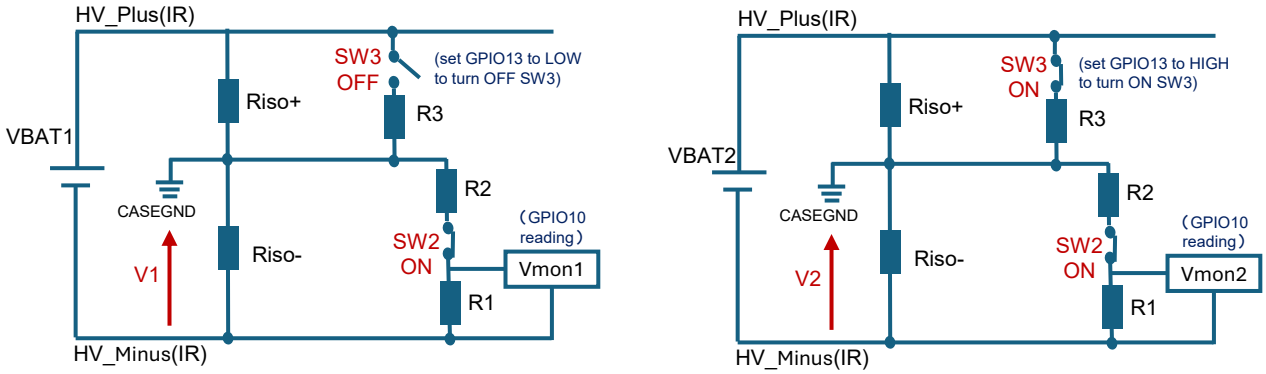


Figure 3.58 Circuit diagrams for Insulation detection function

Known Parameters (Based on the PM-IC Board Circuit):

- **VBAT1 / VBAT2** – Voltage across the HV_Plus (IR) and HV_Minus (IR) terminals on the PM-IC board
- **Vmon1 / Vmon2** – Derived GPIO10 voltage value of PM-IC
- **R1 / R2 / R3** – Resistor parameters determined by the PM-IC board circuit configuration.

General equation used in calculation:

- The equivalent **impedance** is calculated using admittance summation for parallel elements.

$$R=\frac{1}{Y} \qquad R_x \parallel R_y = \left(\frac{1}{R_x}+\frac{1}{R_y}\right)^{-1} = \left(Y_x + Y_y\right)^{-1} = \frac{1}{Y_x + Y_y}$$

- Assumption: **R4** = R1 + R2

Steps of derivation:

1) Set GPIO13 LOW to turn **OFF SW3**, then derive **V1**.

$$V_1 = V_{BAT1} \times \frac{\frac{1}{Y_{iso-}+Y_4}}{\frac{1}{Y_{iso+}}+\frac{1}{Y_{iso-}+Y_4}} = V_{BAT1} \times \frac{Y_{iso+}}{Y_{iso+}+Y_{iso-}+Y_4}$$

2) Rearrange Eq. (1) and denote it as **Eq. (A)** for later use.

Eq. (A)

$$(V_1 - V_{BAT1}) \times Y_{iso+} + V_1 \times Y_{iso-} + V_1 \times Y_4 = 0$$

3) Set GPIO13 HIGH to turn **ON SW3**, then derive **V2**.

$$V_2 = V_{BAT2} \times \frac{\frac{1}{Y_{iso-}+Y_4}}{\frac{1}{Y_{iso+}+Y_3}+\frac{1}{Y_{iso-}+Y_4}} = V_{BAT2} \times \frac{Y_{iso+}+Y_3}{Y_{iso+}+Y_{iso-}+Y_3+Y_4}$$

4) Rearrange Eq. (3) and denote it as **Eq. (B)** for later use.

Eq. (B)

$$(V_2 - V_{BAT2}) \times Y_{iso+} + V_2 \times Y_{iso-} + (V_2 - V_{BAT2}) \times Y_3 + V_2 \times Y_4 = 0$$

Steps of derivation (cont'):

5) Multiply **Eq. (A)** by V2, and multiply **Eq. (B)** by V1, then take their difference.

Eq. (A) x V2 – Eq. (B) x V1

6) Rearrange Eq. (5)

$$(V_1 - V_{BAT1}) \times V_2 \times Y_{iso+} + V_1 \times V_2 \times Y_{iso-} + V_1 \times V_2 \times Y_4 - (V_2 - V_{BAT2}) \times V_1 \times Y_{iso+} - V_2 \times V_1 \times Y_{iso-} - (V_2 - V_{BAT2}) \times V_1 \times Y_3 - V_2 \times V_1 \times Y_4 = 0$$

7) Rearrange Eq. (6)

$$V_1 \times V_2 \times Y_{iso+} - V_{BAT1} \times V_2 \times Y_{iso+} - V_1 \times V_2 \times Y_{iso+} + V_{BAT2} \times V_1 \times Y_{iso+} + V_1 \times V_2 \times Y_{iso-} - V_2 \times V_1 \times Y_{iso-} + V_1 \times V_2 \times Y_4 - V_2 \times V_1 \times Y_4 - (V_2 - V_{BAT2}) \times V_1 \times Y_3 = 0$$

8) Rearrange Eq. (7)

$$-V_{BAT1} \times V_2 \times Y_{iso} + V_{BAT2} \times V_1 \times Y_{iso+} - (V_2 - V_{BAT2}) \times V_1 \times Y_3 = 0$$

9) Rearrange Eq. (8)

$$(V_{BAT2} \times V_1 - V_{BAT1} \times V_2) \times Y_{iso+} - (V_2 - V_{BAT2}) \times V_1 \times Y_3 = 0$$

10) Derive **Yiso+,** and subsequently obtain the insulation resistance **Riso+**

$$Y_{iso+} = \frac{(V_2 - V_{BAT2}) \times V_1 \times Y_3}{(V_{BAT2} \times V_1 - V_{BAT1} \times V_2)} \qquad R_{iso+} = \frac{1}{Y_{iso+}}$$

11) Substitute Eq. (10) into **Eq. (A)** from the previous step

$$(V_1 - V_{BAT1}) \times V_2 \times Y_{iso+} + V_1 \times V_2 \times Y_{iso-} + V_1 \times V_2 \times Y_4 = 0$$

12) Rearrange Eq. (11)

$$V_1 \times V_2 \times Y_{iso-} = -(V_1 - V_{BAT1}) \times V_2 \times Y_{iso+} - V_1 \times V_2 \times Y_4$$

13) Derive **Yiso-**, and subsequently obtain the insulation resistance **Riso-**

$$Y_{iso-} = \frac{-(V_1 - V_{BAT1}) \times Y_{iso+}}{V_1} - Y_4 \qquad R_{iso-} = \frac{1}{Y_{iso-}}$$

3.5.4 Coulomb Counter

The remaining PM-IC functions (**Coulomb Counter**, **VBAT/Relay Diagnosis**, and **VADC**) are accessed through sub-windows. Click the corresponding button to open each sub-window.

Figure 3.59 PM-IC Window – other functions

The Coulomb Counter (CC) function is accessed by clicking the Coulomb Counter button, which opens the CC sub-window. After configuring the required settings, the user can start the CC operation by clicking the CC START button.

During operation, the GUI periodically requests batches of current/voltage measurement data from the MCU, derives the Coulomb Count and Energy values, and displays the results versus time using graphical plots for reference. The CC measurement interval is configurable through registers, and up to 3,000,000 samples can be collected and logged.

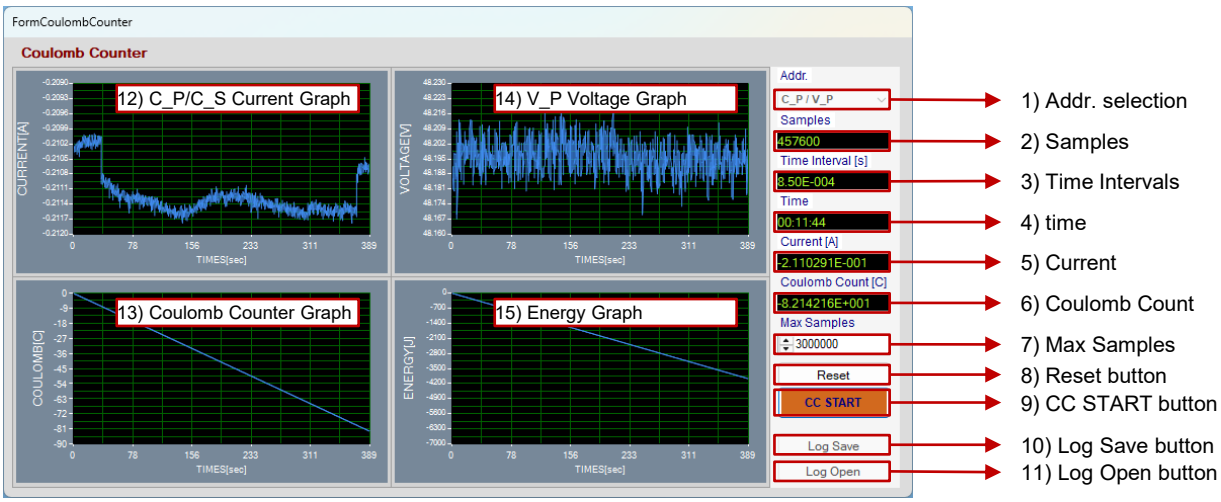


Figure 3.60 PM-IC Window – Coulomb Counter function

1) ***Addr. selection*** – Select the measurement source used for Coulomb counting and data visualization. The selected item will be displayed in the corresponding graph.
   - C_P - Primary current measurement ADC of the PM-IC (Register 0xC7).
   - C_S - Secondary current measurement ADC of the PM-IC (Register 0xC9).
   - C_P / V_P - Combination of primary current (C_P, Register 0xC7) and primary voltage (V_P, Register 0xC8), used for energy calculation.

2) ***Samples*** – Displays the accumulated number of samples received from the MCU. By default, approximately 1300 samples are acquired per acquisition cycle. The sample count is incremented every 1sec a new batch of samples is received.

3) ***Time Interval*** – Displays the ADC measurement interval used for Coulomb counting:
   - When C_P or C_P / V_P is selected, the time interval is defined by register CC_INTVL_P.
   - When C_S is selected, the time interval is defined by register CC_INTVL_S.
   - Default time interval: 850 µs

4) ***Time*** – Displays the accumulated measurement time, starting from 0. The value is updated every 1sec a new batch of samples is received.

5) ***Current*** – Displays the derived current value corresponding to the selected address (C_P or C_S) which is updated every 1sec.

6) ***Coulomb Count*** – Displays the accumulated Coulomb count value. It is calculated by accumulating the product of the measured current and the selected time interval over.

7) ***Max Samples*** – Sets the maximum number of samples used for plotting. (Used to limit graph length and memory usage.)

8) ***Reset button*** – Resets all graphs and measurement settings during the Coulomb Counter operation.

9) ***CC START button*** – Starts or stops the Coulomb Counter function.

10) ***Log Save button*** – Selects the file location before starting the Coulomb Counter operation. After the operation is stopped, the GUI saves the logged selected measurement data, including C_P, C_S, and V_P values. The example logged data csv file is shown as below.

11) ***Log Open button*** – Opens and displays previously saved log files.

12) ***C_P/C_S Current Graph*** – Plots the derived primary current (C_P) or secondary current (C_S) ADC values versus time.

13) ***Coulomb Counter Graph*** – Plots the accumulated Coulomb count value versus time.

14) ***V_P Voltage Graph*** – Plots the derived primary voltage (V_P) ADC value versus time.

15) ***Energy Graph*** – Plots the accumulated energy value versus time. Energy is calculated by accumulating the measured power, derived from voltage and current, over the selected time interval.

	A	B	C
1	Total Samples	13650	
2	TotalCoulomb [C]	-1.86E+00	
3	Interval Time[s]	8.50E-04	
4	Last Current[A]	-1.53E-01	
5	Measure Time	0:00:21	
6	Samples	C_P	V_P
7	1	-1.54E-01	4.88E+01
8	2	-1.45E-01	4.88E+01
9	3	-1.34E-01	4.88E+01
10	4	-1.53E-01	4.88E+01
11	5	-1.64E-01	4.88E+01
12	6	-1.69E-01	4.88E+01
13	7	-1.69E-01	4.88E+01
14	8	-1.65E-01	4.88E+01
15	9	-1.65E-01	4.88E+01

Figure 3.61 Example of data log of Coulomb Counter function

3.5.5 VBAT/Relay Diagnosis

This board provides multiple sensing and measurement ports (VBAT, HV MEAS, Charge±, Link±) for external connection detection and voltage monitoring. These ports are used in conjunction with relay switching and GUI threshold comparison during demo and validation.

The GUI continuously monitors the voltage levels on these ports through internal ADC channels. During demo or validation:

- External relays are switched ON or OFF to connect or disconnect the corresponding port pairs.
- The relay action causes a change in the sensed voltage level at the measurement node.
- The GUI compares the measured voltage against predefined thresholds.
- Based on whether the voltage is above or below the threshold, the system determines:
    - Relay closed (connected)
    - Relay open (disconnected)

This mechanism allows reliable detection of external connection status, relay functionality, and port continuity, without using high-voltage or high-current drive signals. Refer to Chapter 2.3.4 for relay connections.

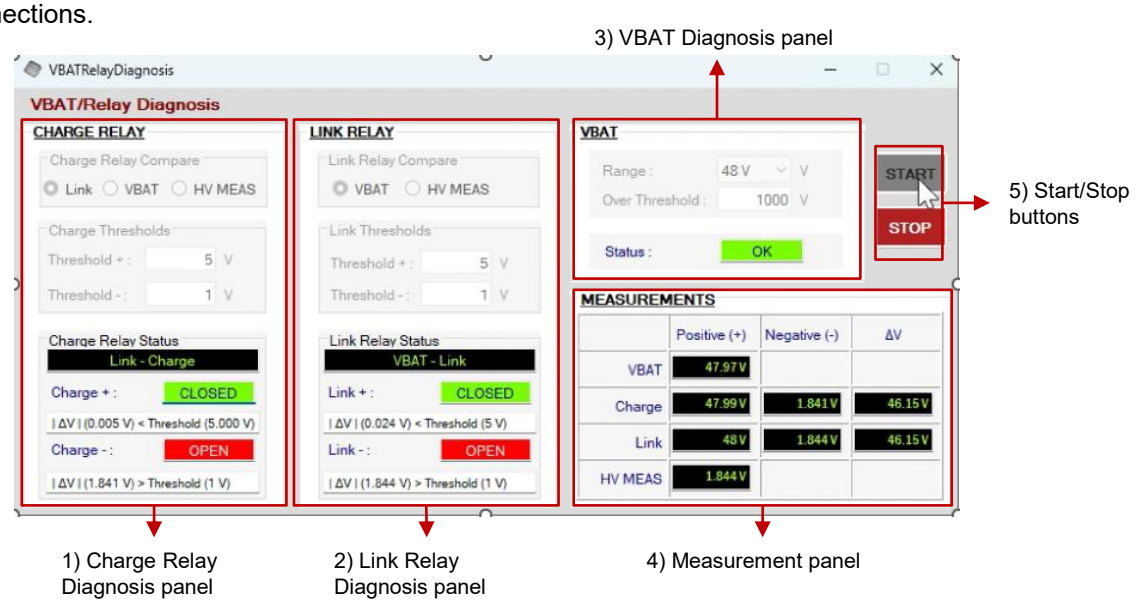


Figure 3.62 PM-IC Window – VBAT/Relay Diagnosis function

The following diagnosis and explanations are based on the measurement results shown in Fig. 3.60.

1) ***Charge Relay Diagnosis panel –*** is used to diagnose the operating status of the Charge Relay by comparing voltage measurements between defined port pairs.

   Port Pair select => e.g.: Link – Charge / VBAT – Charge / HV MEAS – Charge

   Threshold configuration => e.g.: set Charge threshold+ to 5V, Charge threshold- to 1V .

   Diagnosis => e.g.: Charge+ relay closed & Charge- relay open

   Reason: The delta voltage between Link and Charge port measurements is less than the Threshold+ (5 V), and the delta voltage between GND and Charge- port measurements if greater than the Threshold- (1V).

2) ***Link Relay Diagnosis panel –*** is used to diagnose the operating status of the Link Relay by comparing voltage measurements between defined port pairs.

   Port Pair select => e.g.: VBAT – Link / HV MEAS – Link

   Threshold configuration => e.g.: set Link threshold+ to 5V, Link threshold- to 1V .

   Diagnosis => e.g.: Link+ relay closed & Link- relay open

   Reason: The delta voltage between VBAT and Link+ port measurements is less than the Threshold+ (5 V), and the delta voltage between GND and Link- port measurements if greater than the Threshold- (1V).

3) ***VBAT Diagnosis panel –*** is used to diagnose the operating status of the VBAT by comparing voltage measurements between VBAT/HV MEAS port and GND.

   VBAT Range select => e.g.: 48V / 400V / 800V

   Threshold configuration => e.g.: VBAT threshold to 1000V

   Diagnosis => e.g.: VBAT relay status OK

   Reason: The measured voltage at HV MEAS is less than the configured VBAT Threshold+ (1000 V).

4) ***Measurement panel*** – Derive and display the value of each measurement port.

   e.g.: VBAT / Charge+ / Charge- / difference of Charge+ and Charge- / Link+ / Link- / difference of Link+ and Link- / HV MEAS

5) ***Start/Stop buttons*** – Click the Start button to begin the diagnosis after configuration. Measurement data and relay diagnosis results are updated every 1sec. Click the Stop button to terminate the diagnosis.

3.5.6 VADC

The derived measurement ADC values are updated from the PM-IC data registers every 1 second.

The table below summarizes the register addresses for each item.

Item Name	Reg Addr	Item Name	Reg Addr	Item Name	Reg Addr
VDD50	0x6D	**VDAC1**	0x73	**GPIO5**	0x7D
VDDIO	0x6E	**VDAC2**	0x74	**GPIO4**	0x7E
VDD21	0x6F	**VDAC3**	0x75	**GPIO3**	0x7F
VDD18	0x70	**VDAC4**	0x76	**GPIO2**	0x80
VDD21MVSS31	0x71	**VDAC5**	0x77	**GPIO1**	0x81
DIETEMP	0x72	**VDAC6**	0x78		
VBATP	0x7B	**VDAC7**	0x79		
VBATN	0x7C	**VDAC8**	0x7A		

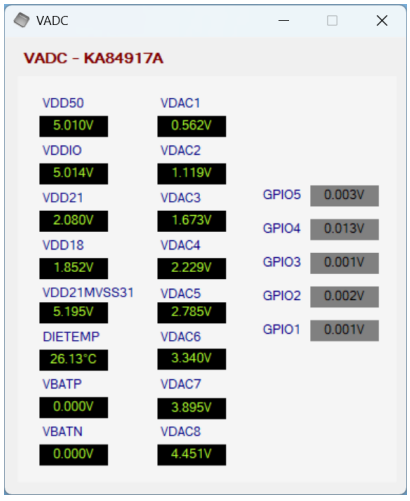


Figure 3.63 PM-IC Window – VADC function

4. LED ALLOCATION AND INDICATION

4.1 LED Definition for Operation Mode and Error Factor

The MCU board features five LEDs, namely LED1 (Red), LED2 (Amber), LED3 (Blue), LED4 (Green), LED5 (Red). LED1~LED4 are used to indicate the LED Mode status, while LED5 is used to indicate power supply of MCU board. The LED Mode status is categorized for operation mode and error factor. Different combinations of LED illumination correspond to specific scenarios. The table below provides a detailed explanation of each LED status.

X: OFF; O: ON; B: Blinking (1Hz)

LED MODE	LED STATUS				EXPLANATION
	4	3	1	2	
System Error	X	O	O	O	System error detected (e.g., BM-IC control or SPI communication error = HIGH).
Standby Mode	X	X	B	B	Operation mode = Standby.
Shutdown Mode	X	B	B	X	Operation mode = Shutdown.
Active Mode – SPI Error	B	O	O	O	Operation mode = Active with SPI error (ERR, CRCERR, TIMOUT, DAISY_ERR).
Cell Balancing	B	B	X	X	Operation mode = Active with cell balancing.
Active (Normal)	B	B	B	B	Operation mode = Active without errors or balancing.
Safety Diagnostic	X	B	B	B	Safety diagnostic mode in progress.
Default	X	X	X	X	Default / System startup error / undefined case.

Table 4.1 LED Definition for Operation Mode and Error Factor

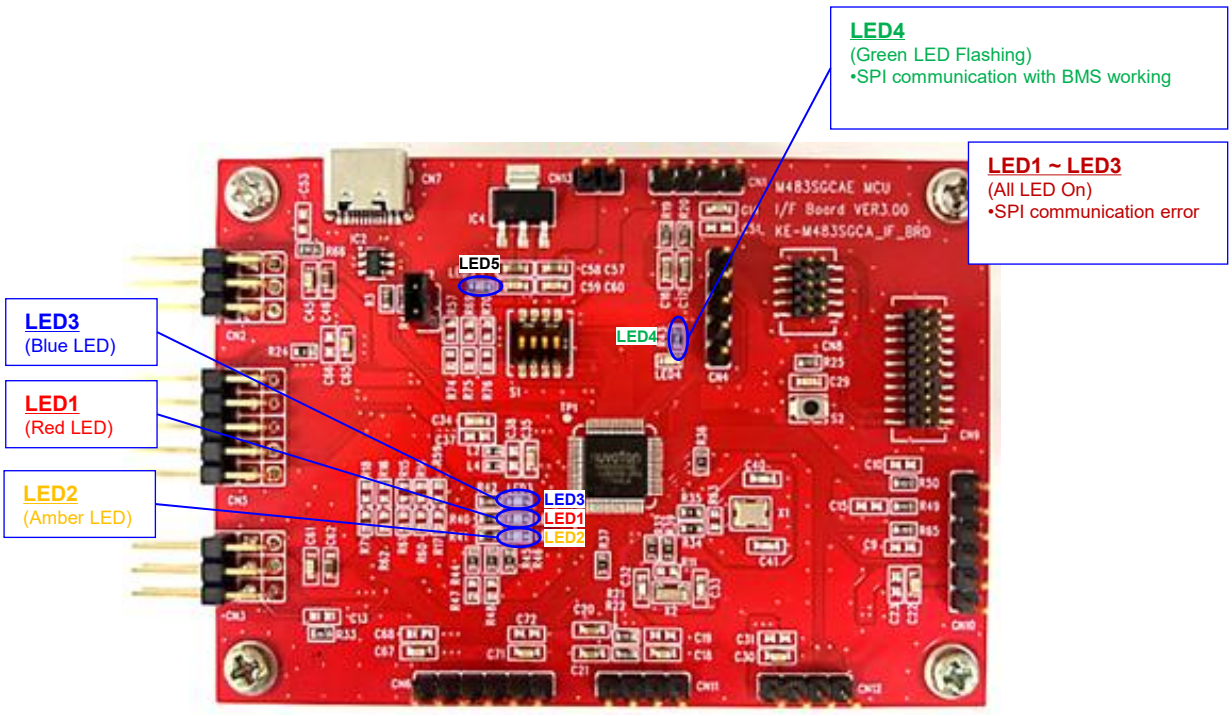


Figure 4.1 LED allocations and indications

4.2 LED Definition for Firmware Update Operation

The MCU board features five LEDs, namely LED1 (Red), LED2 (Amber), LED3 (Blue), LED4 (Green), LED5 (Red). LED1~LED4 are used to indicate the LED Mode status, while LED5 is used to indicate power supply of MCU board. The LED Mode status is categorized for firmware update status. Different combinations of LED illumination correspond to specific scenarios. The table below provides a detailed explanation of each LED status.

X: OFF; O: ON; B: Blinking (2Hz) B*: Blinking (4Hz)

LED MODE	LED STATUS				EXPLANATION
	4	3	1	2	
Update Error	B*	B*	B*	B*	Indicates an error during the update process, such as flash erase/write or CRC mismatch.
Update Complete	O	O	O	O	Firmware update has been completed successfully.
Update In Progress (>75%)	B	O	O	O	Firmware update progress is greater than 75%.
Update In Progress (50% - 75%)	X	B	O	O	Firmware update progress is between 50% and 75%.
Update In Progress (25% - 50%)	X	X	B	O	Firmware update progress is between 25% and 50%.
Update In Progress (<25%)	X	X	X	B	Firmware update progress is less than 25%.
Waiting for Update	X	X	X	B	Waiting for the firmware update to start.

Table 4.2 LED Definition for FW Update status

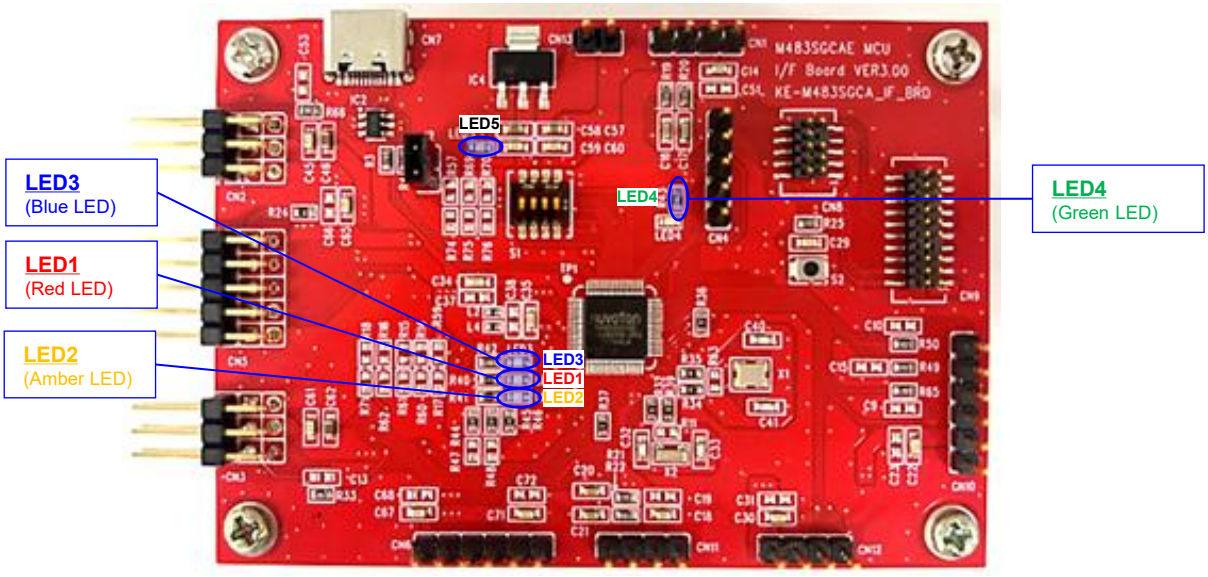


Figure 4.2 LED allocations and indications

REVISION HISTORY

Ver.	Date	Revised Contents
V1.00	10th Oct 2025	1. Initially issued.
V2.00	30th Mar 2026	1. PM-IC contents added
V3.00	30th Jun 2026	1. COMM-IC contents added 2. Fast CV Data Log contents added 3. The maximum number of device connected changed from 11 to 22

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.