RAA489204 Sample Code

This quick start guide provides instructions how to build a test setup with RAA489204 Battery Front End and EK-RA2A1 evaluation kits. It also explains how to install and run the sample project. The quick start sample code is demonstrated using a simple user interface and a terminal software.

Contents

1. Introduction

The Battery Front End (BFE) RAA489204 is an industrial grade Li-ion battery manager IC that supervises up to 14 series connected cells and can operate in standalone mode or in a stack with up to 30 BFEs, monitoring up to 420 Li-ion cells in total. The quick start sample code demonstrates the specially developed RAA489204 software package that provides robust and easy access to the resources and functionality of the BFE. This guide provides instructions for running the quick start sample code. For more details and information about application, configuration and operation of the sample code, see the documents on the RAA489204 page.

1.1 Assumptions and Advisory Notes

- It is assumed that you possess basic understanding of microcontrollers, embedded systems hardware, battery management systems and Li-based battery cells.
- It is assumed that you have prior experience working with Integrated Development Environments (IDEs) such as e^{2}studio and terminal emulation programs such as Tera Term.
- The screen shots provided throughout this document are for reference. The actual screen content may differ depending on the version of software and development tools used.
- Renesas recommends reviewing the *EK-RA2A1 Quick Start Guide*, *EK-RA2A1 Manual, RAA489204 Datasheet,* and *RTKA489204DK0000BU Evaluation Kit Manual* to get acquainted with MCU and BFE features before proceeding further.

1.2 Overview of the Quick Start Sample Project

The quick start sample project includes the following components:

- Hardware Abstraction Layer – The Flexible Software Package (FSP) drivers that provide access to the peripherals and resources of the selected MCU.
- Battery Front End Abstraction Layer – The middleware Application Programming Interface (API) for connecting the BFE control functions with an upper-level application and the implementations of all API functions providing access to the full functionalities of RAA489204. The communication drivers with the device-specific SPI protocol are also available.
- Demo Application – A simple demonstration of the basic API functions that includes a data visualization software, fully-functional cell balancing algorithm, and fault management example.

The state machine flow diagram of the sample project is illustrated on Figure 1. It shows the relations between states and diagrams and the conditions for transition. A state executes its function and moves to the next state or mode, while the mode can remain static or loop inside until a transition flag is set. The execution of the code begins with Initialization State where the MCU initializes a Battery Front End interface and runs a setup and self-test. Then, it stays in Idle Mode where the sample application is waiting for a user input from the command line interface and runs memory test every 10 cycles and full self-test every 1000 cycles. The Balancing Mode contains a cell balancing algorithm and exit automatically when the process is finished. Fault State can be entered from any other state or mode as soon as an error code is returned or the FAULT pin is asserted. Keep in mind that the quick start sample project does not handle any charge/discharge FET control or current measurement.

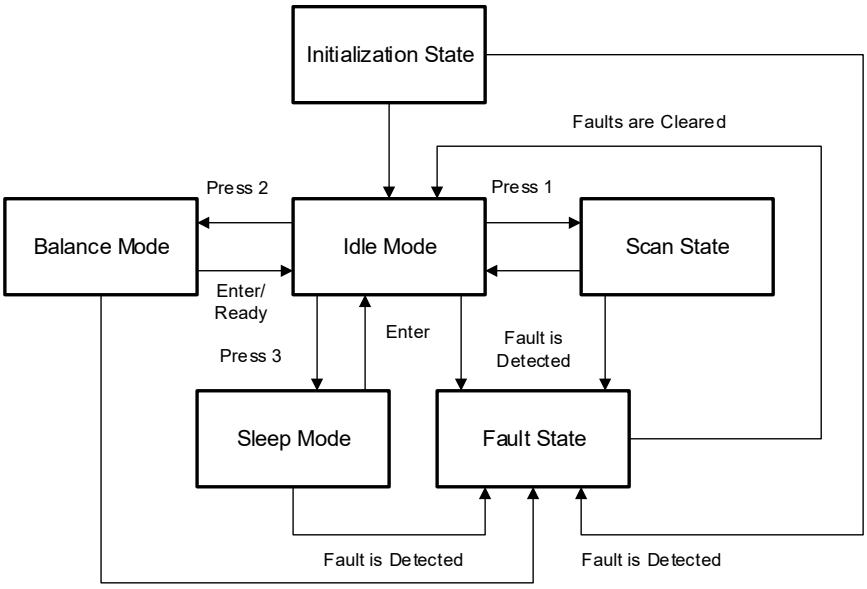

Figure 1. Sample Code State Machine Flow Diagram

2. Running the Quick Start Example Project

2.1 Hardware Requirements

The hardware setup, needed for running the demo and familiarization with the software package, is given in Table 1. It contains evaluation kits, PC or notebook, power supply, and measurement and control tools. Some of the items are considered as optional. The used MCU evaluation board does not provide any galvanic insulation. Therefore, additional isolation may be required when working with high voltage battery packs (not needed in the proposed demonstration setup). It also can be changed with EK-RA4W1 or another from Renesas RA MCU family. However, this process requires modifications of the sample project. For more information, refer to the software manuals, available on the product page. A combination of a DC power supply and resistor ladders emulate the battery pack. At a typical operating voltage of 49V result in 3.5V per cell. The stack of three BFE eval kits consume about ~140mA in normal operation mode and ~112mA in sleep mode. **Important:** Do not go under 12V or exceed 65V. A precision multimeter is used as a voltmeter to monitor battery pack voltage, individual cell voltages, or another voltage of interest. The recommended start-up connection is between Pack+ (J5) and Pack- (J6) (MCB_PS4_Z). An oscilloscope can be used optionally to capture signals in the SPI communication channel between the MCU and stack master device together with the FAULT and DATA READY signals. The vertical daisy chain communication ports can be captured when using a differential probe or two passive probes, referenced to ground and subtracted using the scope math tool. For easier capture and better results, Renesas recommends an oscilloscope equipped with a logic analyzer.

Table 1. Hardware Setup

#	Item	Qty.	Description
1	RTKA489204DE0000BU	3[1]	RAA489204 Evaluation Board
2	MCB_PS4_Z	3[1]	14 Cell resistor ladder board (or target battery pack)
3	EK-RA2A1	1	RA2A1 Evaluation Board
4	USB micro cable	2	Connection between EK-RA2A1 and PC
5	USB isolator (optional)	1	Dual isolator. Additional 5V power supply might be required
6	DC Power Supply	1	6VDC to 60VDC Regulated
7	Precision multimeter	1	Multipurpose DC voltmeter compatible with total battery voltage
8	Oscilloscope (optional)	1	Equipped with Logic Analyzer
9	Cables and wires	-	Connection between MCU and AFE boards, Resistor ladders and power supply
10	Personal computer or notebook	1	Running Windows® or Linux with USB support

1. The quantity depends on the stack size.

2.2 Software Requirements

The required tools for directly compiling, debugging, and running the project provided with the BFE software package under scope, include e^2 studio Integrated Development Environment (IDE) with the GCC Arm® Embedded toolchain, the Flexible Software Package (FSP), and a terminal emulator program. Visit e^2 studio for instructions and more information. If an alternative microcontroller and/or integrated development environment and toolchain are used, you should install them according to related instructions. Next, port the software library together with the demo application and connect them to the hardware abstraction layer API for the selected hardware. Finally, you can run the sample code and go through the example.

2.3 Importing the Sample Project

The sample project provided with this document can be imported into the e^2 studio workspace by following these steps:

1. Right click in the **Project Explorer** view and select **File > Import** from the menu showing up to open the Import Wizard. Ensure that the RAA489204 sample project archive file is already downloaded from the Renesas website.

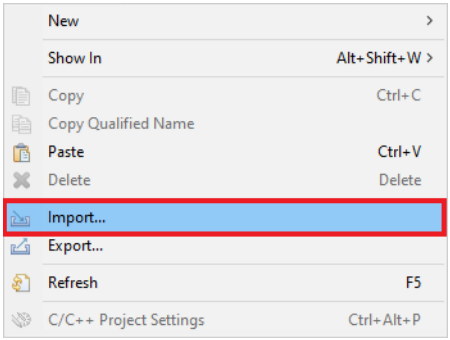

Figure 2. Right-click Menu to Import the Sample Project

2. From the Import Wizard window select **Existing Project into Workspace** or **Rename & Import Existing C/C++ Project into Workspace** and then click the **Next >** button.

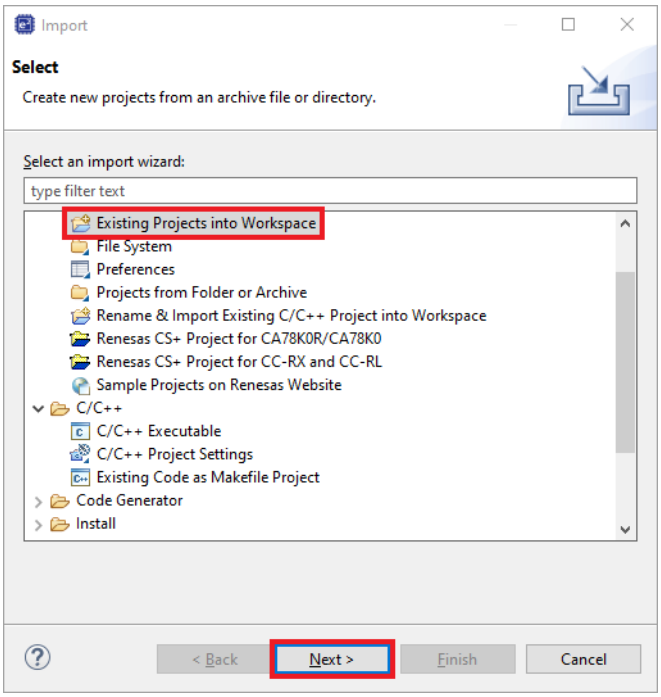

Figure 3. Selection of the Import Wizard Option

3. On the next window select **Select archive file** option and then click the **Browse…** button. Navigate to the location of the sample project archive file (.zip), select it and then click the **Open** button. Click the **Finish** button to complete the import process and close the Import Wizard.

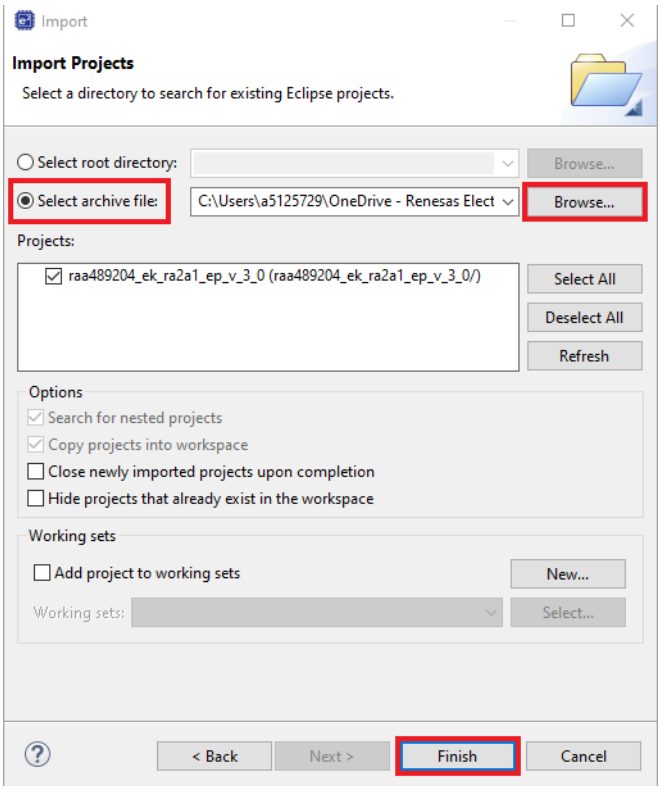

Figure 4. Selection of the Sample Project

4. The project is now imported into the e^2 studio workspace. Figure 5 shows the imported project structure. Double-click on **configuration.xml** to open the FSP Configurator and press **Generate Project Content** button to generate the hardware specific project files, needed to run the project on the selected hardware. Now the project is ready to be built.

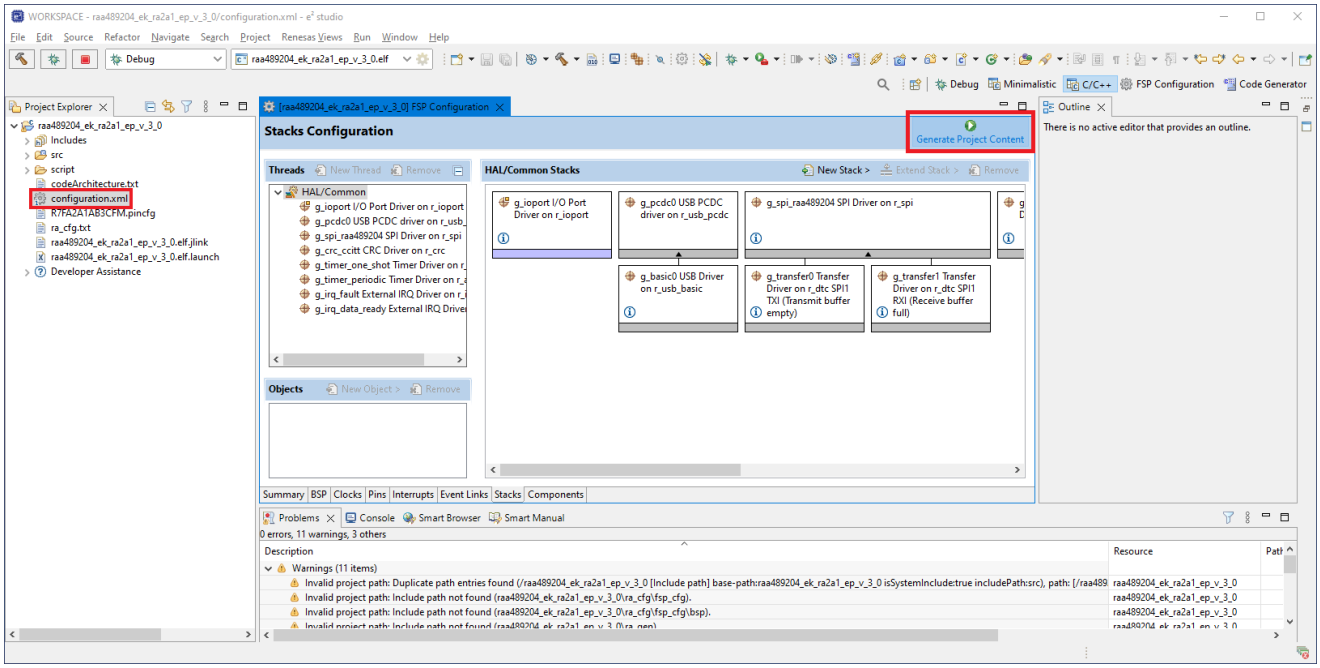

Figure 5. Generation of Additional Project Files

5. Open **Project** from top menu and press **Build All** button. Then open **Run** from top menu and select **Debug As > 3 Renesas GDB Hardware Debugging**. The IDE changes automatically to a debug perspective and a debug session is started (Figure 6). Press the **Resume** button twice to run the code.

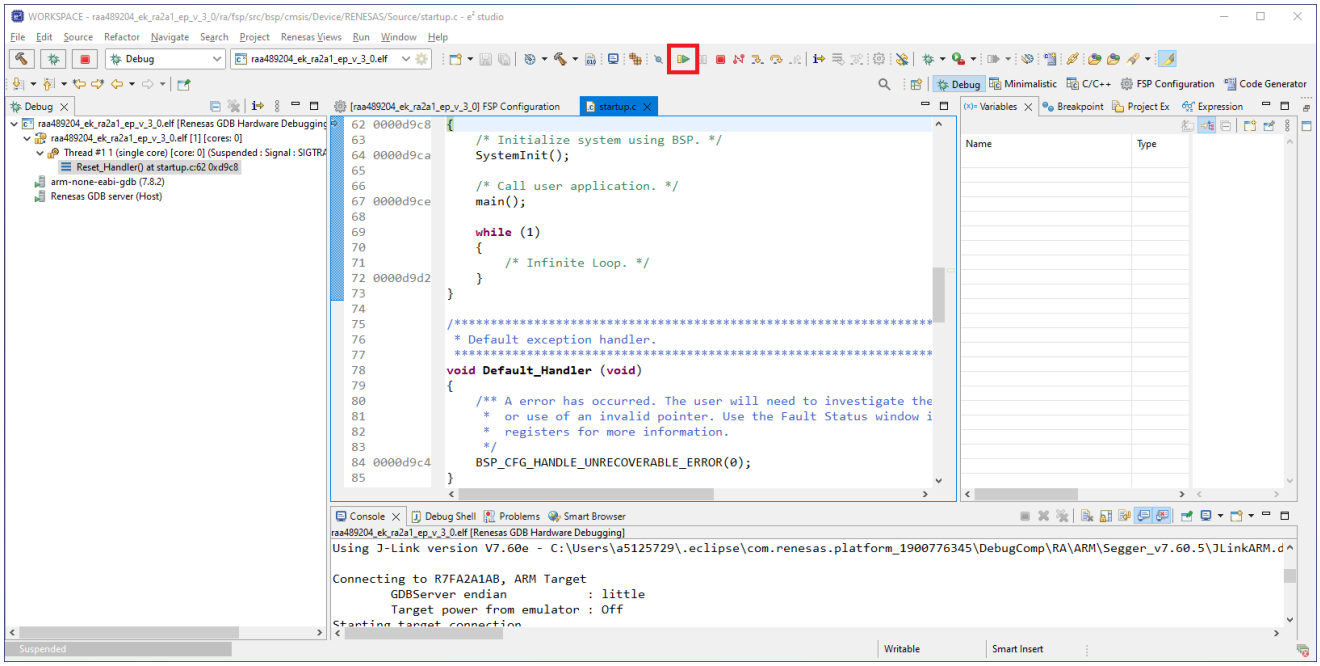

Figure 6. Generation of Additional Project Files

2.4 Connecting and Powering up the Setup

The hardware assembly for a single RAA489204 board is shown in Figure 7, and for a stack of three BFEs is shown in Figure 8. When stacked, the RAA489204 boards are interconnected directly using headers JP5 and JP6. Nevertheless, twisted pair cables or the specialized crimped with RJ45 connector cables for transformer coupling can also be used. The MCU and RAA489204 boards must be connected with external jumper cables (see Table 2). On the resistor ladder board, ensure that JP2 is placed for 14-cell position (OFF) and on the EVK, ensure that the LEDEN jumper is in place and all other jumpers J2, J5, J6, J10 are in the correct position as shown on the figures. For more details, please, refer to the RTKA489204DK0000BU Evaluation Kit Manual.

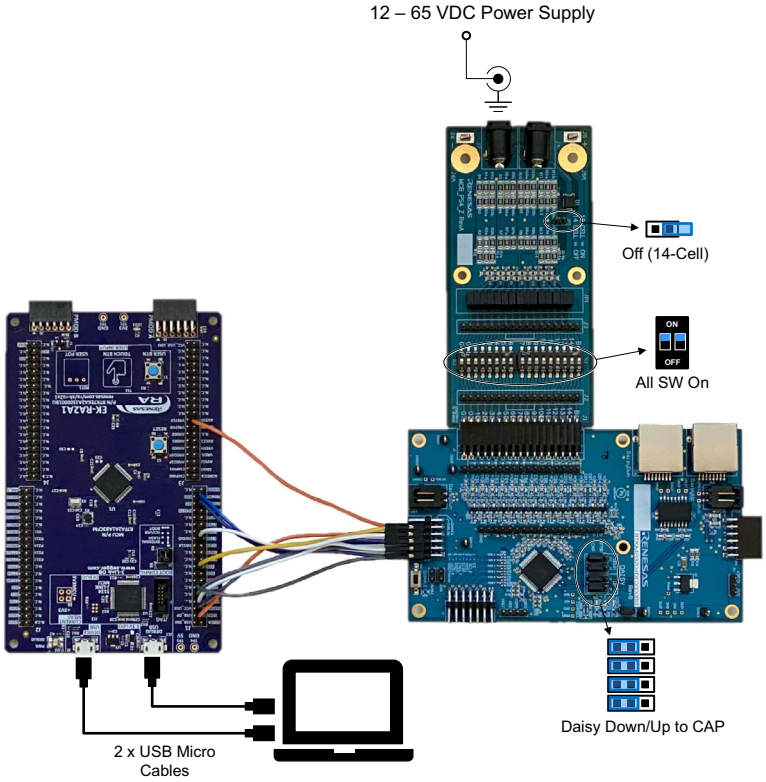

Figure 7. Connecting the MCU and Standalone RAA489204 Eval Boards

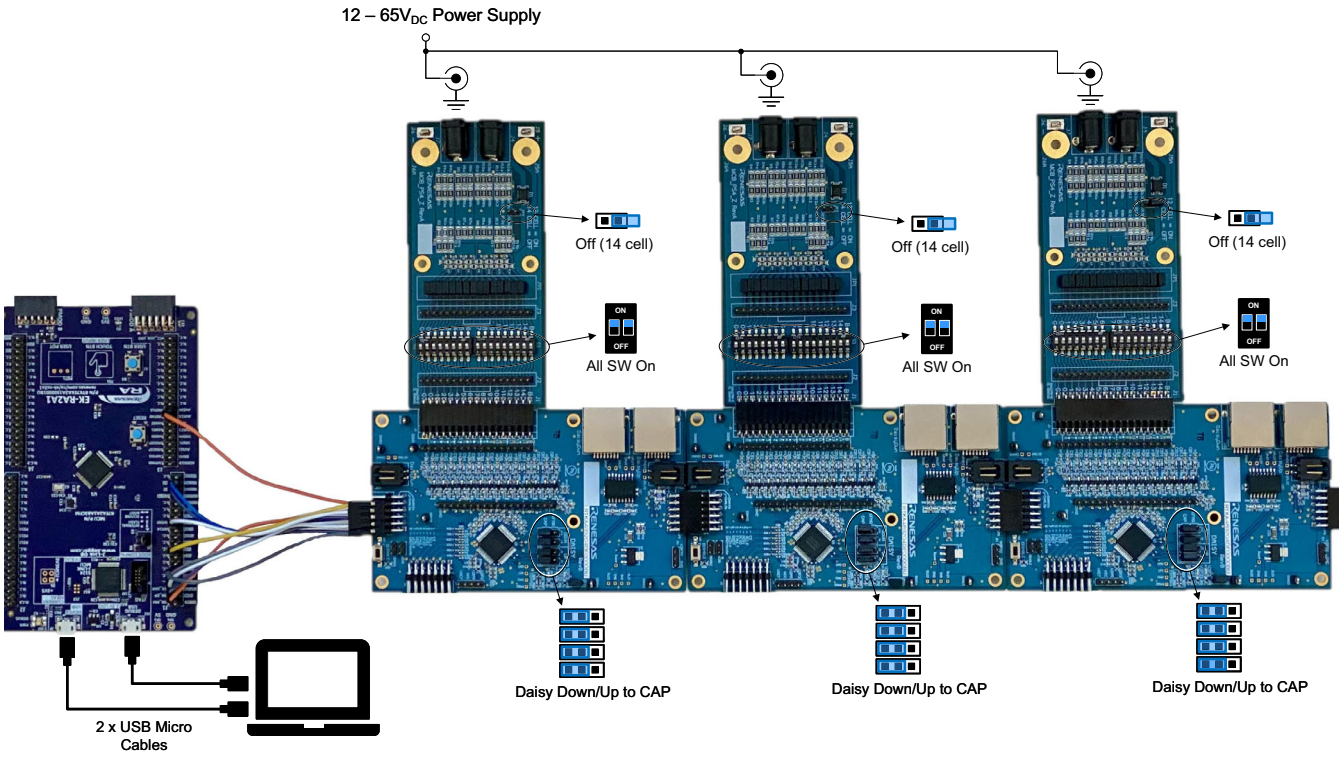

Figure 8. Connecting the MCU and a Stack of Three RAA489204 Eval Boards

Important:

- Before connecting the EK-RA2A1 evaluation board, ensure all power sources are off and the USB cables are unplugged.
- Ensure the negative terminal of the voltage source powering the resistor ladder is grounded.
- Ensure the EVK board is grounded when reconnecting the USB cables; otherwise, EK-RA2A1 Eval Board can be damaged by grounding currents because galvanic insulation is not available.

The MCU board is powered by the host computer through the debug USB connector J11. To enable the virtual serial communication channel with the terminal software, the device USB J9 must be connected. The resistor ladders are connected and powered up in parallel so that no matter how many boards are stacked, the supply voltage is always up to 65V for safety reasons. The positive terminal of the MCB_PS2_Z board is J5/J5A, and the negative terminal is J6/J6A. Connectors J4 and J7 can also be used to power to the board with nonregulated 24V power supplies. The J4/J7 tip is positive and the ring is negative.

Table 2. BFE and MCU Board Connection

Signal name	R7FA2A1AB3CFM - Port	EK-RA2A1 - Header	RTKA489204DE0000BU (master) - Header
CS	P102	J1 - SSLB0	JP6 – 1 (CS)
SCLK	P103	J1 - RSPCKB	JP6 – 2 (SCLK)
DIN	P105	J1 - MOSIB	JP6 – 3 (MOSI)
DOUT	P104	J1 - MISOB	JP6 – 4 (MISO)
FAULT	P110	J1 - GTIOC2B	JP6 – 5 (FAULT)
DREADY	P109	J1 - GTETRG	JP6 – 6 (DRDY)
EN	P402	J1 - CTS9	JP6 – 7 (uCDIS)
DTRDYMode	P401	J1 - SDA0	JP6 – 8 (DRDYMode)

Table 2. BFE and MCU Board Connection (Cont.)

Signal name	R7FA2A1AB3CFM - Port	EK-RA2A1 - Header	RTKA489204DE0000BU (master) - Header
GND	--	J1 - VSS	JP6 – 9 (GND)
CMode	--	J3 – AVSS0	JP6 – 10 (CMode)

Before turning on the setup, ensure the BMS sample project is imported into e^2 studio IDE (See Importing the Sample Project).

The sample code is preconfigured to correspond to the hardware setup. The default settings are used directly with the items from Table 1 (stack of three Battery Front Ends). If any modification is made, review all settings in **r_bms_cfg.h**, **r_bfe_cfg.h**, and **bal_data.c** (refer to the software manuals from the product page). If real lithium battery cells are used instead of resistor ladders, the undervoltage and overvoltage limits and balancing constants must be carefully set corresponding to the used chemistry and cell balancing requirements.

The hardware setup must be turned on in the following sequence:

1. Ensure that everything is properly connected and both USB cables are connecting the MCU evaluation board and the PC. Check the jumpers and switches on all boards.
2. Set the power supply to 49V and power up the resistor ladder (MCB_PS4_Z board). Check that the voltages on the J3 connector of the board are +3.5V from cell to cell.
3. In e^2 studio, build the sample project and initiate debugging to download the software to the MCU. Run the software. Ensure that the time between powering up the resistor ladder and running the sample code is less than 60s. Otherwise, the BFE enters Sleep Mode, and you must go back to Step 2.
4. Open the Tera Term terminal and press **Enter** to display the menu.

The MCU should start with or after the BFE; otherwise, it returns an error when trying to initialize an unpowered RAA489204 Battery Front-End.

2.5 Use of the Command Line Interface

A terminal emulator program is needed for running the user interface (UI). A useful option is Tera Term, which can be downloaded from the Tera Term Home Page. To initiate a terminal session, a new connection must be opened (Figure 9). The serial connection settings for the terminal software are given in Table 3. The PC user must have read/write access permission for the USB port.

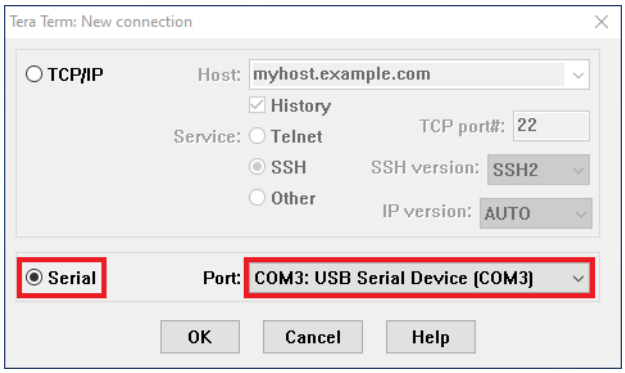

Figure 9. Tera Term New Connection Window

If Windows® recognized the board correctly, it is listed in Tera Term as a serial connection. If the board is not listed at all or the Device Manager indicates an error, there might be a problem with the driver. See the latest support entry for this topic in the Renesas Knowledge Base to resolve this.

Table 3. Serial Connection (Terminal) Settings

Parameter	Value
New Line (Receive)	CR
New Line (Transmit)	CR
Terminal Mode	VT100
Baud Rate	115200
Data Bits	8 bits
Parity	none
Stop Bits	1 bit
Flow Control	none

When the hardware setup is fully connected and powered-up, and the terminal session is initiated, the user interface becomes active. Press the **Enter** key and the MCU displays the selection menu shown in Figure 10. Now the BMS is in Idle Mode and the MCU is waiting for a user input. By pressing **1**, the BMS goes to Scan State all voltages, temperatures, and GPIOs are measured. Results are read and displayed onto the terminal within a table.

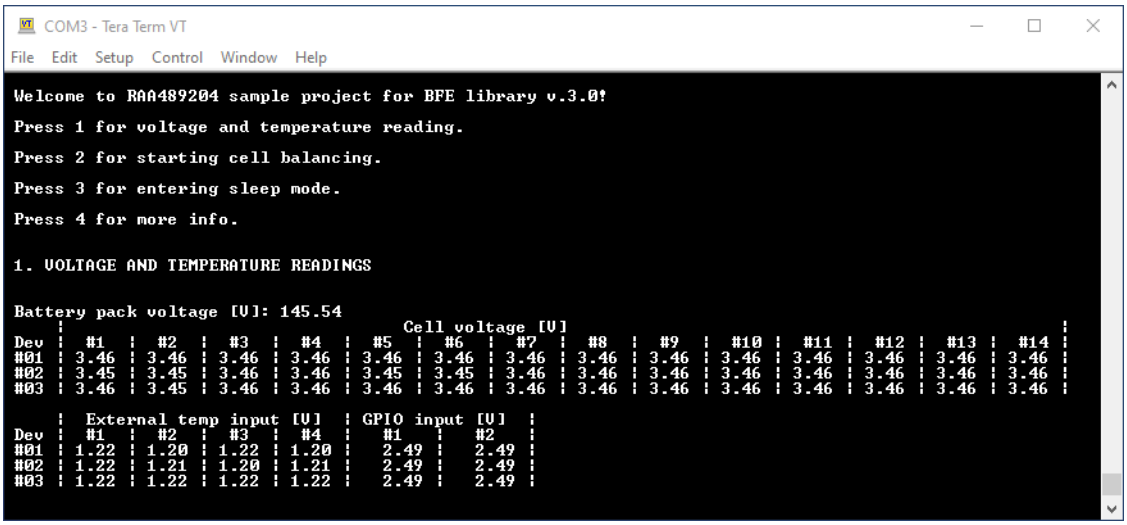

Figure 10. User Interface - Voltage and1 Temperature Readings

When you press **2** inside the menu, the BMS goes to Balancing Mode (Figure 11) and the balancing algorithm immediately starts. After each loop, an asterisk is displayed. When all cells are balanced, the MCU sends a message and BMS goes back to Idle State. The balancing activity can be preliminary interrupted by pressing **Enter**. The parameters related to cell balancing cannot be modified trough the command line interface. They are hardcoded into the sample project configuration file (**r_bms_cfg.h**). If you initiate cell balancing with the resistor ladder or if the maximal voltage difference between cells is less than 20mV by default, the BMS enters and then immediately exits Balancing Mode.

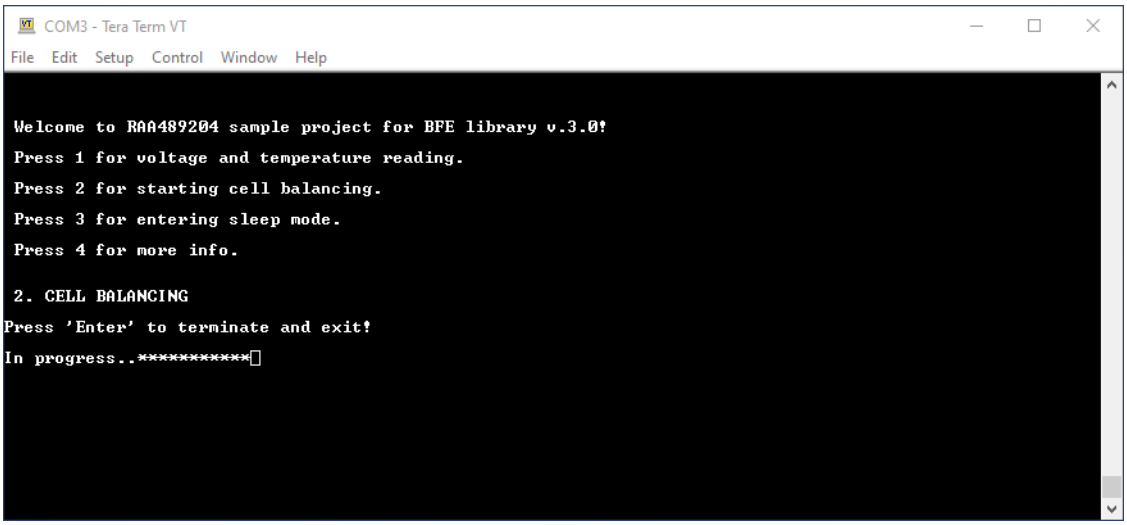

Figure 11. User Interface – Cell Balancing

When you press **3** inside the menu, the BMS goes to Sleep Mode. This can be noticed as the V3P3X LEDs of the RAA489204 eval boards turn off. If a watchdog timeout is used, it expires despite the Sleep Mode, asserting the FAULT pin is asserted, and the Fault LED is on. When the system is in Sleep Mode, it can wake up and return to Idle Mode by pressing **Enter**. If a fault condition is detected, the system goes automatically into Fault State, and the command line interface displays the reason for that and the available options (Figure 12). The Fault LED of the MCU board is on. You can run the built-in fault management algorithms by pressing **Enter**. If the faults are successfully cleared, the BMS goes to Idle Mode; otherwise, a message is displayed that a hard reset is required.

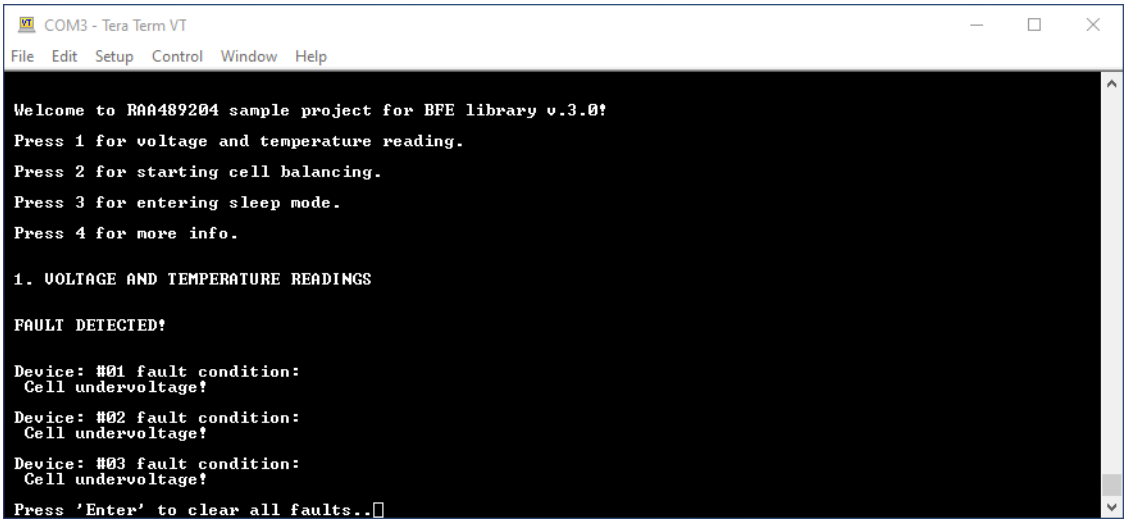

Figure 12. User Interface – Fault Detection

3. Revision History

Revision	Date	Description
1.00	Jul 18, 2022	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/