RAA489220 BFE Sample Code

This quick start guide provides instructions for building a test setup using the RAA489220 Battery Front End (BFE) evaluation kit and the RA4W1 PMOD dongle, and it includes procedures for installing and running the sample project. The quick start sample code is demonstrated using a simple user interface and terminal software.

Contents

1.   Introduction

The Battery Front End (BFE) RAA489220 is an industrial grade Li-ion battery manager IC that monitors and protects 10 Li-based battery cells, two thermistors, and both charge and discharge currents. Without the presence of an MCU, the BFE is capable of an autonomous operation that automatically controls the charging and discharging process and recovers from certain fault conditions. The BFE requires a reduced number of external components such as RC filters, a stunt resistor, power FETs, up to two optional NTC-based temperature measurement networks, and two additional components.

The quick start sample code demonstrates the RAA489220 software package; the package provides robust and easy access to the resources and functionality of the BFE. This quick start 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 Renesas website for RAA489220 documentation.

1.1   Assumptions and Advisory Notes

▪ You should possess basic understanding of microcontrollers, embedded systems hardware, battery management systems and Li-based battery cells.

▪ You should have prior experience working with Integrated Development Environments (IDEs) such as e2studio and terminal emulation programs such as Tera Term.

▪ The screen shots provided throughout this document are for reference. The actual screen content might differ depending on your version of software and development tools.

▪ Before proceeding, Renesas recommends reviewing the *RAA489220 Datasheet*, *RTKA489220DK0000BU Evaluation Kit Manual* and the other application notes and manuals to get acquainted with the BFE features. See the website for RAA489220 documentation.

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 RAA489220. The communication drivers with the device-specific I^2C protocol are also available.

▪ Demo application: A simple demonstration of the basic API functions that includes a data visualization software, auto-scan, and an FETs control functionalities example.

The state machine flow diagram of the sample project is illustrated in 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 mode can remain static or loop inside until a transition flag is set. The execution of the code begins with the Initialization State where the MCU initializes a BFE interface and runs a setup and self-test. Next, it stays in Idle Mode as the sample application is waiting for a user input from the command line interface while running a full self-test every 50 cycles (or approximately every 5s). The Charge and Discharge FETs can be manually toggled when the BMS is in Idle Mode. In Scan State, the BFE runs measurement of all voltages, currents, and temperatures before returning to idle. In Auto Scan Mode, the BFE takes control of the FETs and fault monitoring. A Fault State can be entered from any other state or mode as soon as an error code is returned or the ALERT pin is asserted.
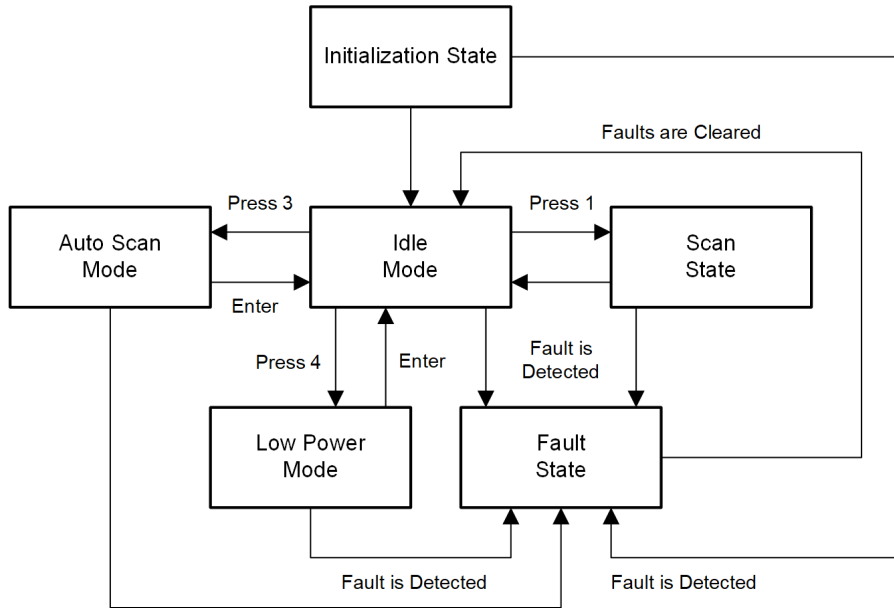
Figure 1. Sample Code State Machine Flow Diagram

2.   Running the Quick Start Example Project

2.1   Hardware requirements

The hardware setup required for running the demo and using the software package (see Table 1) contains evaluation kits, PC or notebook, on-chip debugging emulator, power supplies, and measurement and control tools. Some of the items are optional. Also, verify that the RA4W1 PMOD dongle does not provide any galvanic insulation, and the low-side FETs are disconnecting the negative terminal of the battery pack. A combination of a DC power supply and a resistor ladder emulates the battery pack at a typical operating voltage of a 36V result in 3.6V per cell. The evaluation kit consumes about ~37mA in normal operation mode.

IMPORTANT: Do not go under 6V or exceed 44V.

A precision multimeter could be used as a voltmeter to monitor battery pack voltage, individual cell voltages, or another voltage of interest. The start-up connection is between 10CELL and GND (BMS_PS_CELL10Z). An oscilloscope can be used optionally to capture signals in the I^{2}C communication channel between the MCU and BFE with the ALERT and RESET signals. For easier capture and better results, Renesas recommends an oscilloscope equipped with a logic analyzer.

Table 1. Hardware Setup

#	Item	Qty.	Description
1	RTKA489220DE0000BU	1	RAA489220 Evaluation Board
2	BMS_PS_CELL10Z	1	10-Cell resistor ladder board (or target battery pack)
3	RTKAPMODDONGLEEVZ1	1	RA2A1 Evaluation Board
4	Renesas E2 (Lite) Emulator	1	On-chip debugging emulator
5	USB micro cable	1	Connection between RA4W1 PMOD dongle and PC
6	USB isolator (optional)	1	Additional 5V power supply might be required
7	DC power supply	1	6 - 45V$_{DC}$, regulated
8	Personal computer or notebook	1	Running Windows® or Linux with USB support
9	Precision multimeter (optional)	1	Multipurpose DC voltmeter compatible with total battery pack voltage

Table 1. Hardware Setup

10	DC current source (optional)	2	$0 - 250mA_{DC}$, regulated
11	Oscilloscope (optional)	1	Equipped with Logic Analyzer

2.2   Software requirements

For directly compiling, debugging, and running the project, required tools are provided with the BFE software package under scope and include the following: the e2 studio Integrated Development Environment (IDE) with the GCC Arm® Embedded toolchain, the Flexible Software Package (FSP), and a terminal emulator program. Visit e2 studio for instructions and more information. If an alternative microcontroller and/or integrated development environment and toolchain are required, you should install them according to related instructions, port the software library together with the demo application, and connect them to the hardware abstraction layer API for the selected hardware; this process allows you 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 e2 studio workspace by following these steps:

1. To open the Import Wizard, right-click in the Project Explorer view and select **File > Import** from the menu that appears (Figure 2). *Note*: Verify that the RAA489220 sample project archive file has been downloaded from the Renesas website.
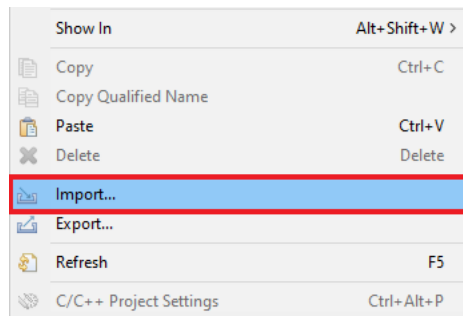

Figure 2. Right-click Menu to Import the Sample Project

2. From the Import Wizard window (Figure 3), select **Existing Project into Workspace** or **Rename & Import Existing C/C++ Project into Workspace** and then click the **Next >** button.

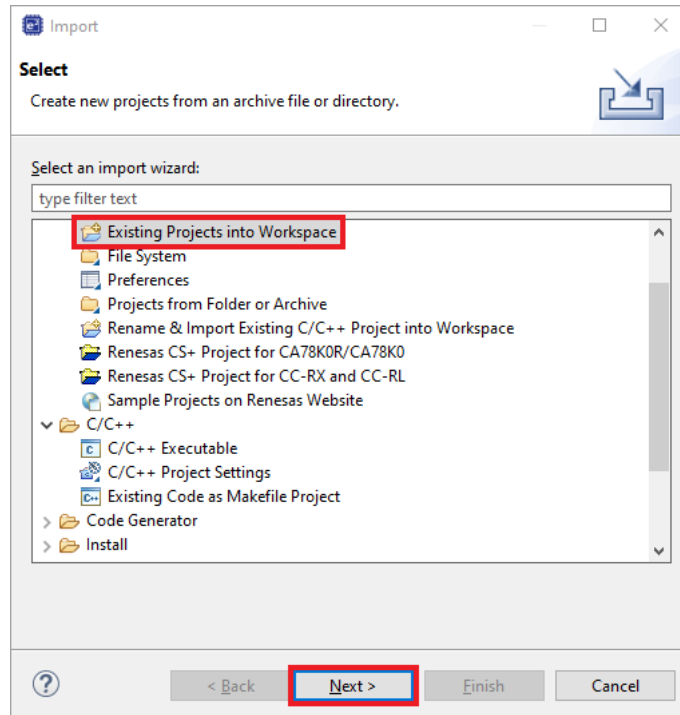**Figure 3. Selection of the Import Wizard Option**

3. In the Import Projects window, choose **Select archive file**, and click the **Browse…** button. File Explorer appears. Navigate to the location of the sample project archive file (.zip), select it, and click the **Open** button. Click the **Finish** button to complete the import process and close the Import Wizard.
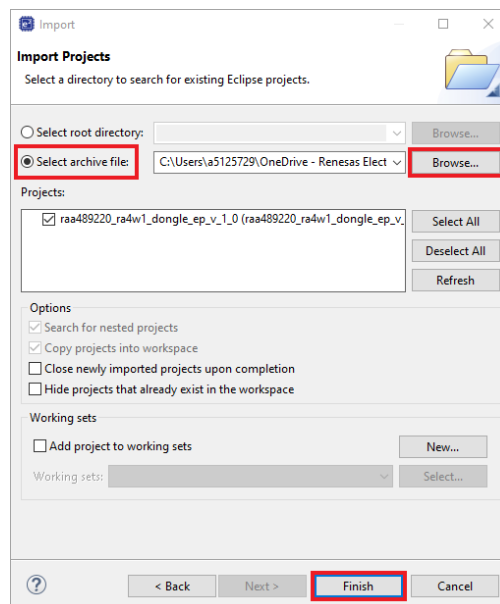

Figure 4. Selection of the Sample Project

4. The project is imported into the e2 studio workspace, and Figure 5 shows the imported project structure. Double-click on **configuration.xml** to open the FSP Configurator. Click the **Generate Project Content** button to generate the hardware specific project files that are needed to run the project on the selected hardware. The project is ready to be built.
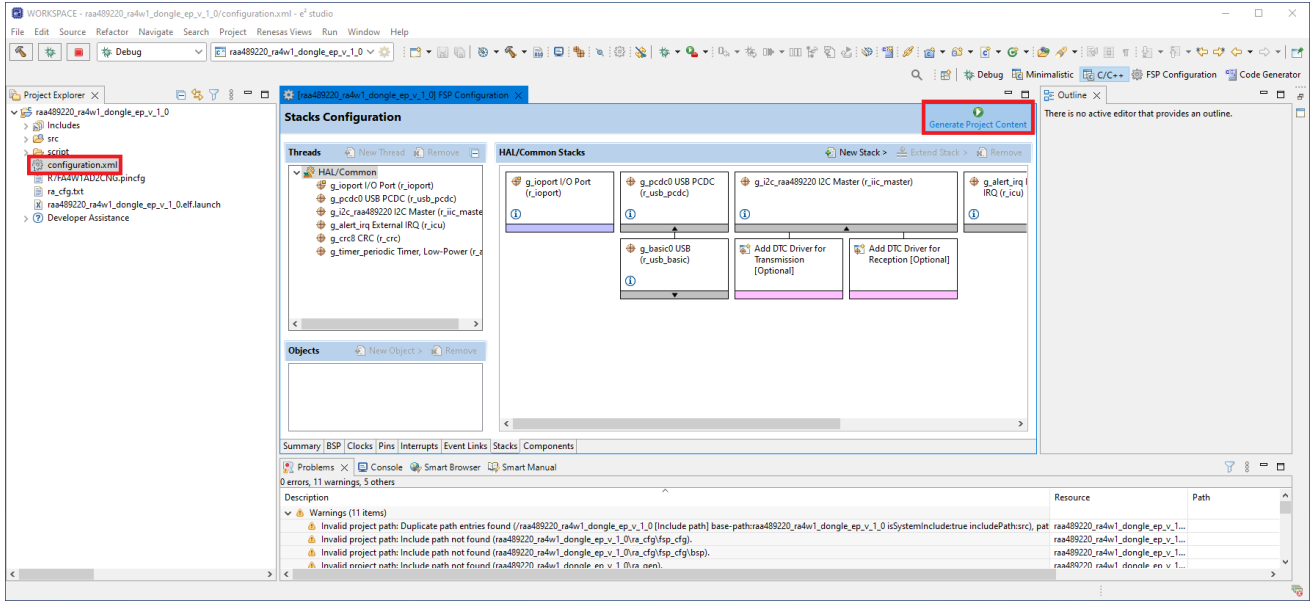
Figure 5. Generation of Additional Project Files

5. From the top menu, choose your project and select the **Build All** button. Next, 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). Click the **Resume** button twice to run the code.
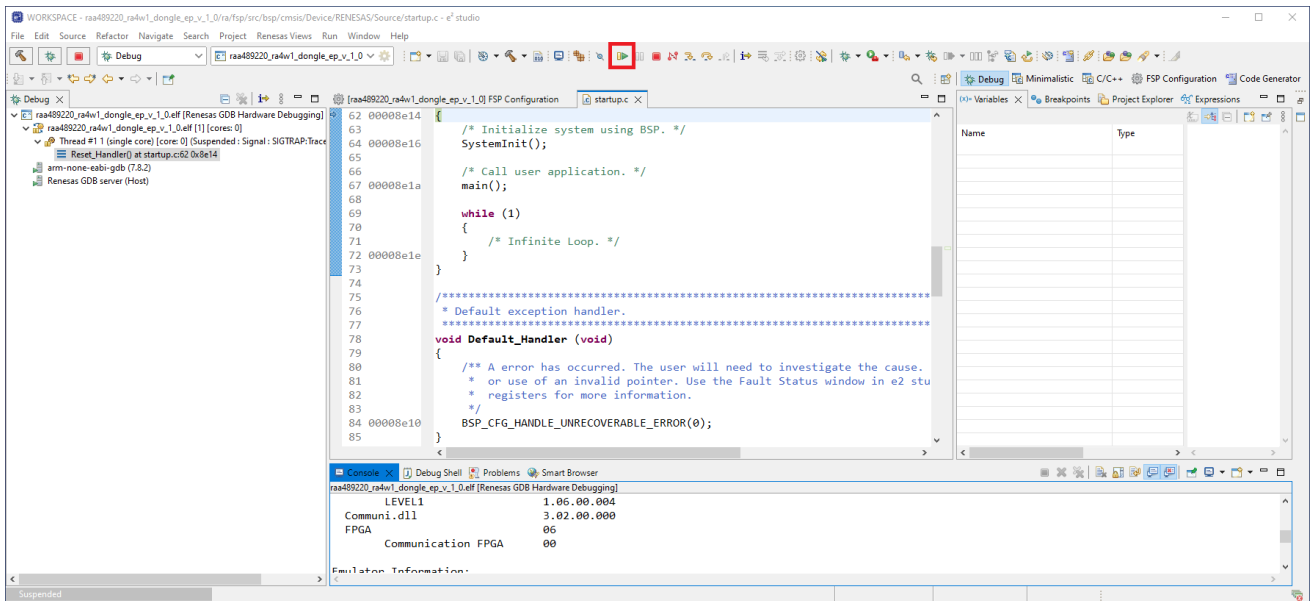

Figure 6. Generation of Additional Project Files

2.4   Connecting and Powering up the Setup

The hardware assembly for a single RAA489220 board is shown in Figure 7. Connect the resistor ladder BMS_PS_CELL10Z, used as a battery emulator, to the RAA489220 EVK. Connect the grounded regulated DC voltage source between GND and 10CELL terminals. Ensure to place the jumper J1 on RTKA489220DK0000BU. Attach an optional current source between J4 and J8 or J8 and J5 to emulate discharge or charge currents respectively. The current direction must be kept as demonstrated in Figure 7. Connect the RA4W1 PMOD dongle. Verify that jumpers on the dongle are on the following positions: JB1(OPEN), JB2(1,2), J3(CLOSED),

J4(CLOSED), J14(OPEN), JB3(2,3), JB4(OPEN), JB7(1,2), JB8(2,3), JB9(OPEN), JB10(1,2), and J14(OPEN). Connect the on-chip debugger to J11.Confirm that the ribbon cable approaches the JTAG connector from the left side as shown in Figure 7. Finally, connect the on-chip debugger and the dongle to the PC using USB cables. For more details, refer to the *RTKA489220DK0000BU Evaluation Kit Manual*.

IMPORTANT:

▪ Before connecting, ensure all power sources are off and the USB cables are unplugged.

▪ You must ground the negative terminal of the voltage source powering the resistor ladder.

▪ Ensure the EVK board is grounded when you reconnect the USB cables; otherwise, the RA7W1 PMOD dongle or the E2 (Lite) Emulator could be damaged by grounding currents, as there is not any galvanic insulation available.
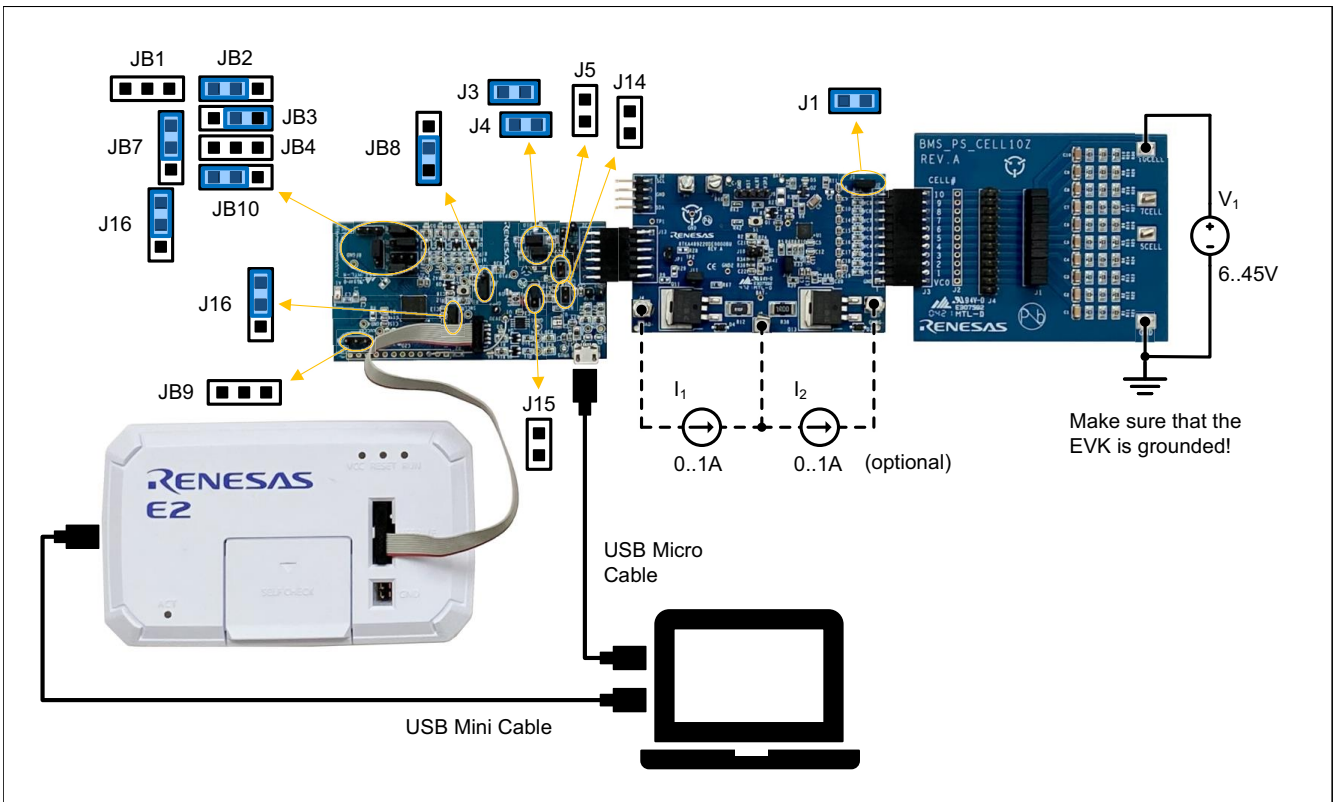
Figure 7. Connecting and Configuring the Dongle, Debugger, and the RAA489220 EVB

Before turning on the setup, ensure the BMS sample project is imported into e2 studio IDE (See Importing the Sample Project). The sample code is preconfigured to correspond to the hardware setup. The default settings are set so that it can be used directly with the items from Table 1. 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 for RAA489220 on the Renesas website). If real lithium battery cells are used instead of a resistor ladder, the undervoltage and overvoltage limits must be carefully set, corresponding to the used chemistry and the charge/discharge current capabilities.

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

1. Ensure that everything is properly connected, the micro USB cable is connecting the RA4W1 PMOD dongle and the PC, and the mini USB cable is connecting the E2 Emulator Debugger and the PC. Check jumpers of all boards.

2. Set the power supply to 36V and power up the resistor ladder (BMS_PS_CELL10Z board). Check that the voltages on the J4 connector of the board are +3.6V from cell to cell.

3. In e2 studio, build the sample project and initiate debugging to download the software to the MCU. Run the software.

4. Limit the discharge current source to 100mA, 1V and enable its output (optional).

5. Limit the charge current source to 10mA, 1V and enable its output (optional).

6. Open the Tera Term terminal and press **Enter** to display the menu.

The MCU should start together with or after the BFE; otherwise, it returns an error when trying to initialize an unpowered RAA489220 BFE.

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 website. To initiate a terminal session, a new connection must be opened (Figure 8). The serial connection settings for the terminal software are given in Table 2. The PC user must have read/write access permission for the USB port.

Figure 8. 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 could be a problem with the driver. See the latest support entry for this topic in the Renesas Knowledge Base to resolve this.

Table 2. 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 connected and powered-up and the terminal session is initiated, the user interface becomes active. Press the **Enter** key so that MCU displays the selection menu (as shown in Figure 9). The BMS

is in IDLE MODE, and the MCU waits for a user input. By pressing **1**, the BMS goes to Scan State all voltages, and the currents and temperatures are measured. Results are read and displayed onto the terminal in a table.
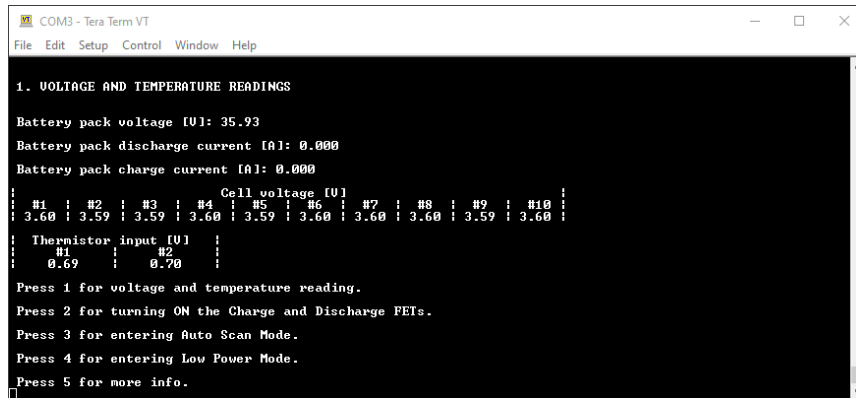

Figure 9. User Interface – Voltages, Currents, and Temperatures Readings

The second option in the menu provides control over the FETs. When you press **2** inside the menu, both charge and discharge FET states are toggled. Pressing **3** enables the Auto Scan Mode. The BFE is preforming a Single System Scan every 100ms, and the data are read and displayed every 5s (Figure 10); this can be terminated by pressing **Enter**.
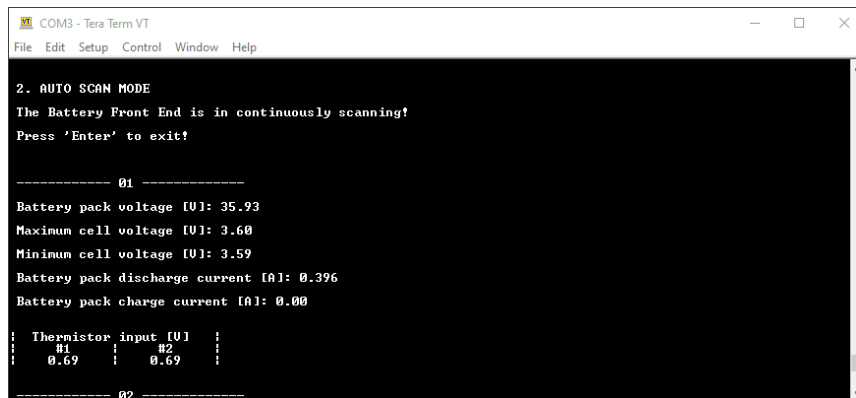
Figure 10. User Interface – Auto Scan Mode

When you press **4** inside the menu (Figure 11), the BMS goes to Low Power Mode. The V3P3 voltage remains stable, but the power consumption of the BFE drops significantly. Keep in mind that Low Power Mode can be hardly confirmed only by monitoring the total EVK consumption as the resistor ladder prevails. You can measure the V_{PACK} current with an Ampere meter in the microamps range. However, long cables can add inductance and result in self-test failure, so keep the cables as short as possible.
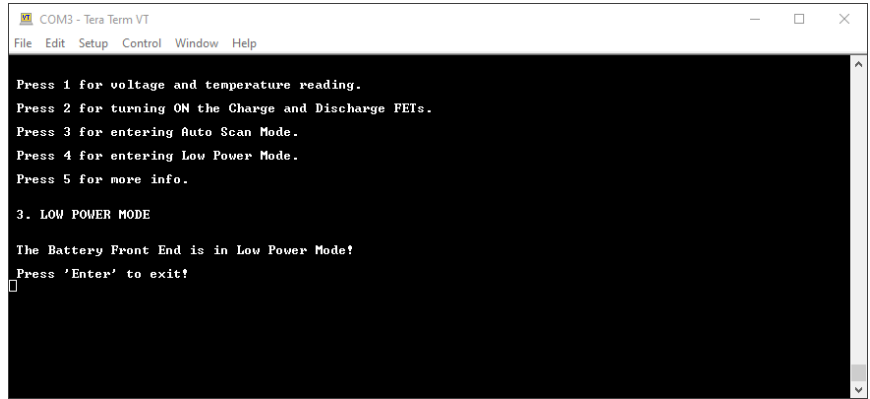
Figure 11. User Interface – Low Power Mode

When the system is in Low Power Mode, it can wake up and return to Idle Mode by pressing Enter. In Idle Mode, a self-diagnostic is called every 5s. If a fault condition is detected during that diagnostic or by calling any other function, the system goes automatically into Fault Mode, and the command line interface displays the reason and the available options (Figure 12). The fault indicating LED - D7 of the RA4W1 PMOD dongle board is illuminated. You can run the built-in fault management algorithms and try to resolve and clear the fault by pressing Enter. If the faults are successfully cleared, the BMS goes to IDLE MODE; otherwise, a message is displayed that a manual hard reset is required, repeating the steps from the turn-on procedure.
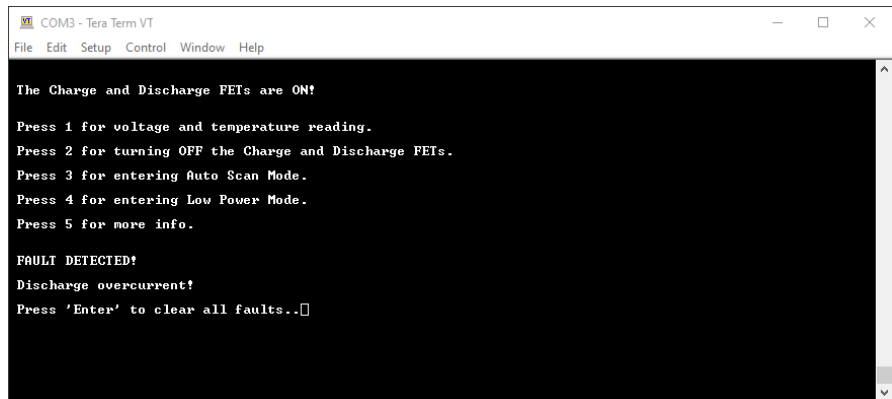
Figure 12. User Interface – Fault Detection

3.   Revision History

Revision	Date	Description
1.00	Aug 16, 2022	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

Corporate Headquarters

Trademarks

Contact Information