Quick Start Guide for RAA489206 16-Cell Battery Front End

This quick start guide reviews the sample project **bfe_raa489206_ek_ra4w1** with the command-line interface (CLI) while demonstrating the features of the RAA4889206 Battery Front End (BFE) and its interactions with an MCU.

Contents

1. Overview

Figure 1 shows the operating environment of the demo project bfe_raa849206_ek_ra4w described in this document, which runs on the EK-RA4W1 board. The BFE board and the attached battery cells can be the RTKA489206DE0000BU evaluation and resistor ladder boards, or any custom board that includes connectivity between the BFE device and the target MCU. The project implements a command-line interface (CLI), which is accessed by a terminal emulator, such as Tera Term on a PC connecting with the EF-RA4W1 board using an USB cable.

The CLI provides commands, which execute the interaction sequences that systems and devices interfacing with BFEs follow to use BFE features. This sample code also contains a sample BMS application that monitors the status of the BFE and reports critical fault events over the terminal interface.

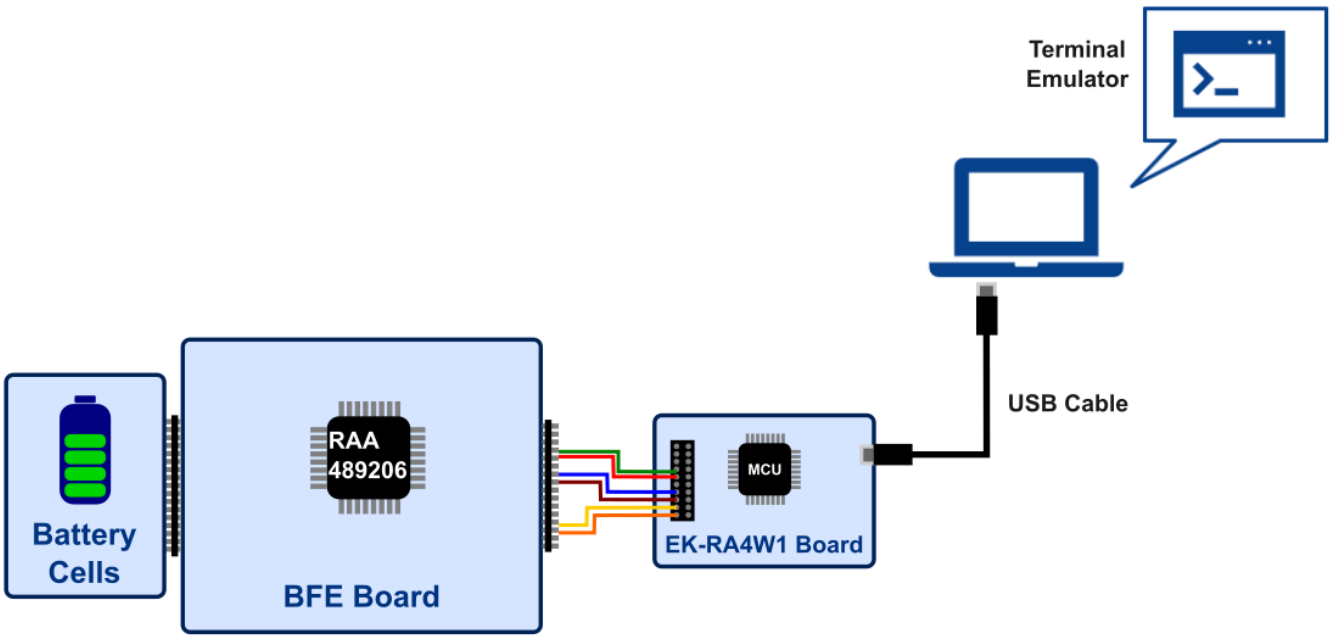

Figure 1. Demo Project Operating Environment

1.1 Assumptions and Advisory Notes

1. It is assumed you possess basic understanding of microcontrollers and embedded systems hardware.
2. Renesas recommends reviewing the *EK-RA4W1 Quick Start Guide* and *EK-RA4W1 Manual*, in addition to the RAA489206 Datasheet and Evaluation Kit Manual, to get acquainted with MCU and BFE features before proceeding further.
3. Flexible Software Package (FSP) and Integrated Development Environment (IDE) such as e2 studio are required to modify, extend, or develop embedded applications on the target EK-RA4W1 kit.
4. Instructions to download and install software, import example projects, build them and program the EK-RA4W1 board are provided in *Renesas e2studio 2021-07 or Higher User's Manual: Quick Start Guide*.

Note: Do not install the sample code into your product. The operation of sample code is not guaranteed. Confirming the operation is your own responsibility.

2. General Software Structure

Figure 2 shows the software structure of the sample code described in this document. The user application code block consists of two modules: the CLI and the sample BMS. The CLI provides commands to interact with the BFE and execute tasks such as:

- Set and read BFE registers using hexadecimal notation.
- Read fault and status indicators, and measurements, such as current, voltage and temperature.
- Set protection thresholds, such as overvoltages and undervoltages, maximum voltage difference between cells, internal over-temperature, and discharge, charge and short-circuit currents.
- Clear faults reported by the BFE.
- Read and set BFE mode.
- Perform continuous scan operation to monitor the battery pack, in addition to single system scans.
- Turn ON and OFF power FET drivers for charge and discharge.

The sample BMS can be started by the CLI. It is a sample application that uses the continuous scan operation feature BFE to monitor and protect the battery pack, typical functions of BMSs.

Both CLI and sample BMS applications interact with the BFE through the BFE Abstraction Layer (BAL). The BAL defines a BFE Interface as a structure consisting of an Application Program Interface (API), a Control Structure, and a Configuration Structure. The BAL works as a middleware between the user application code and the hardware. It decouples user applications code from the software that drives the direct interaction with the BFE and allows usage of BFE features through the API of the BFE interface module. Whereas BFE interface structures (API, Control, and Configuration) are mainly declarations of BFE features, the RAA489206 Instantiation defines and implements the interactions that provide those features. The instance uses the Hardware Abstraction Layer (HAL) of Renesas Flexible Software Package (FSP) to access and use MCU peripherals and modules.

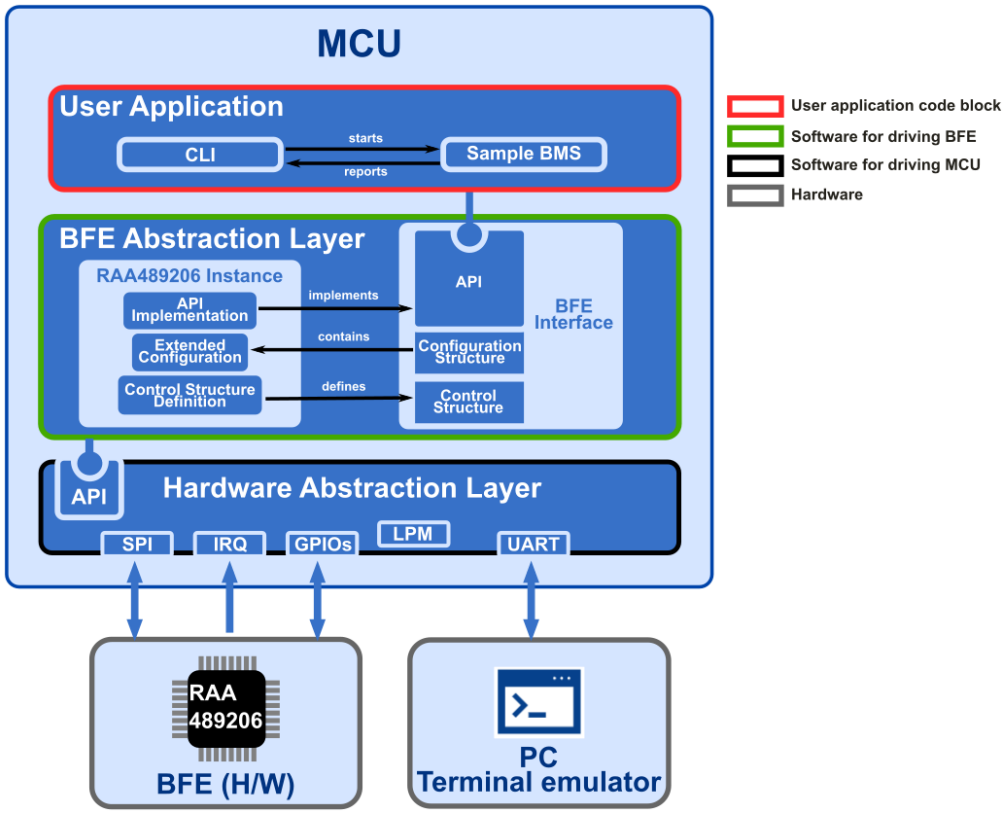

Figure 2. Software Structure of the Sample Code

The BFE instance uses the following APIs of HAL to interact with the BFE device:

- External Interruption Request (IRQ) Interface to detect the ALERT pin events generated by the BFE.
- Serial Peripheral Interface (SPI) Interface to communicate with the BFE.
- General Purpose Input/Output (GPIO) to access and configure I/O ports that configure the communication interface and reset the BFE.
- Universal Asynchronous Receiver-Transmitter (UART) to communicate with the terminal emulator.

Low Power Mode (LPM) to control the power consumption of the MCU during the execution of applications.

Table 1 shows the structure of the sample code. The modules shown in bold within the gray cells contain the code related to the use of BFE functionalities; their code can be modified to extend BFE features or adapt to the requirements of the intended case.

Table 1. Directory Structure of the Sample Code

Directory				Description	Module
ra	fsp	inc	api	Modules APIs	HAL (Generated by FSP)
			instances	Definition of modules instances	
		src	r_*.c	APIs Implementations	
ra_gen	---			Instantiation of HAL modules and main.c that calls the entry point	
ra_cfg	fsp_cfg	r_*_cfg.h		Configuration options files	
src	hal_entry.c	---		Entry point that calls the application main	
	bfe	**---**	**r_bfe_api.h**	**BAL API**	**BAL**
			r_bfe_raa489206_cfg.h	**Configuration macros**	
			r_bfe_raa489206.*	**BFE instance and API implementation**	
	app_lib	**cli**	***.c** ***.h**	**Command-Line Interface implementation**	**User Application**
		cmd	***.c** ***.h**	**CLI commands**	
		user_app	**r_bms.***	**BMS sample application**	
			r_cli_main.*	**CLI application main called by entry function in hat_entry.c**	
			r_coulomb_counting.*	**Coulomb Counting functions**	
			r_icr1865026j_02a.*	**Lookup table values of released capacity vs. open circuit voltage of the ICR18650_26J cell**	
			r_lookup_table.*	**Function that looks and interpolates values in a lookup table**	
			r_soc.*	**State-of-Charge (SOC) application**	

For details on code architecture and implementation of the API to drive, interact with and feature RAA489206 BFE functionalities, see the Application Note *MCU Sample Code for Driving the RAA489206 16-Cell Battery Front End*.

3. How to Use the Demo Project

This section describes the procedure to import the demo project that contains the sample code.

3.1 Operating Environment

Table 2 and Table 3 show the hardware and software requirements to build and debug the provided sample software.

Table 2. Hardware Requirements

Hardware	Description
Host PC	Windows® 10 PC with USB interface
MCU Board	EK-RA4W1 (RTK7EKA4W1S00000BJ)
On-chip debugging emulator	The EK-RA4W1 has a J-Link on-board debugger, so no external debugger is necessary
USB cables	Two USB A/USB micro B cables to connect the EK-RA4W1 (Debugger and serial) to the PC

Table 3. Software Requirements

Software		Version	Description
GCC environment	e2 studio	2022-04	Windows® 10 PC with USB interface
	GCC ARM Embedded	10.2.1.20201103	C/C++ compiler (download available from e^2 studio installer)
	Renesas Flexible Software Package (FSP)	3.3.0 or higher	Software package for development of projects with the Renesas RA series of MCU devices
	Segger J-Flash	V6.94	Tool to program on-chip flash memories of MCU devices
Header files			All API calls and their supporting interface definitions located in the header files (*.h) contained in the src directory
Integer types			ANSI C99 Exact width integer types declared in stdint.h

3.2 Importing the Demo Project

The Demo project provided with this document can be imported into an e2 studio workspace by completing the following steps:Select **File** > **Import**

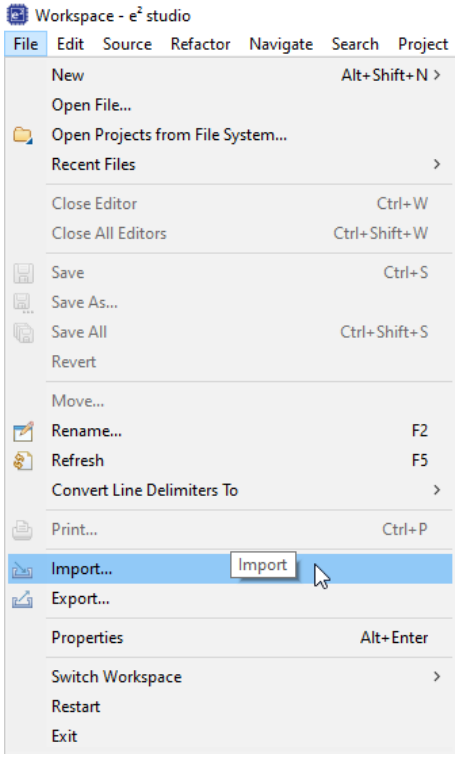

Figure 3. File Menu to Import the Demo Project

5. Select **Existing Project into Workspace** and click **Next** button.

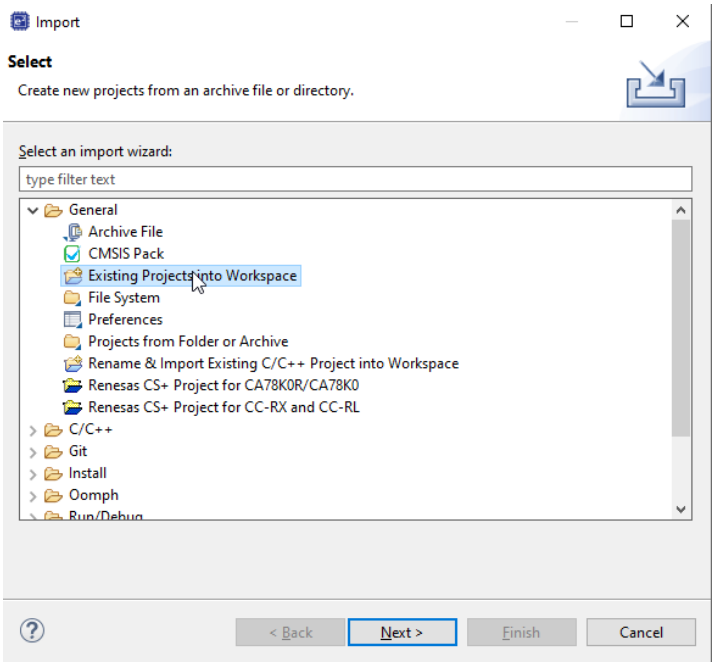

Figure 4. Selection of the Import Option

6. Select the **Select archive file** option, click the **Browse**… button and then select the demo project file (.zip). Click the **Finish** button.

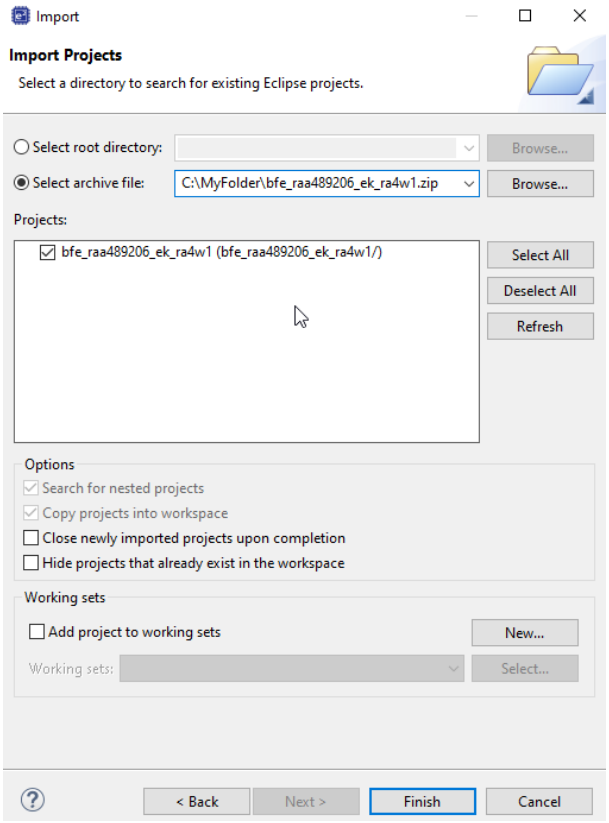

Figure 5. Import the Sample Project

7. The project is now imported into the e2 studio workspace. Figure 6 shows the imported project structure

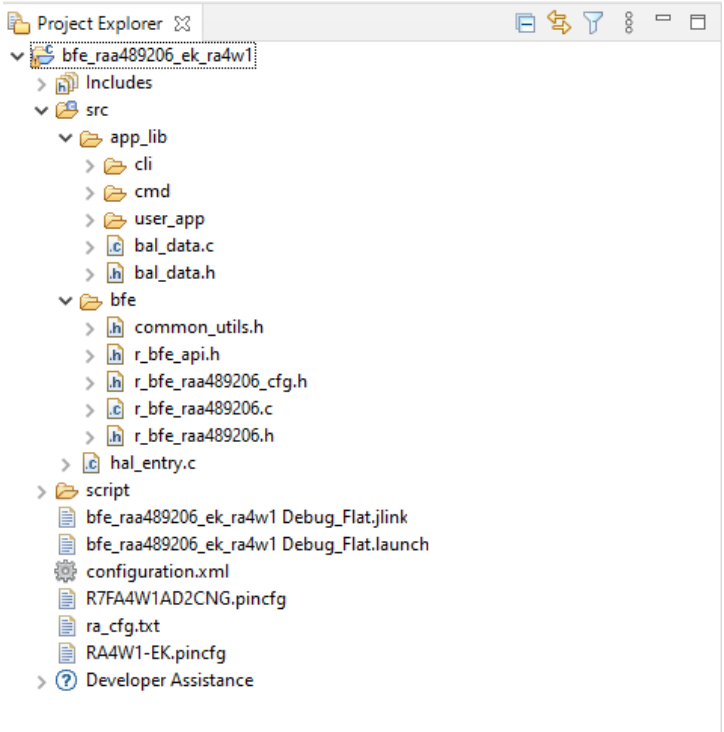

Figure 6. Structure of the Sample Project

3.3 Building and Debugging

Reference the *Renesas e2studio 2021-07 or Higher - User's Manual: Quick Start Guide.*

3.4 Demo Project Functional Description

3.4.1 BFE and EK-RA4W1 Boards

The sample project requires the RA489206 BFE device to be properly mounted on a board with the required circuitry as specified by its datasheet. It is also necessary that the BFE board allows direct connectivity between the EK-RA4W1 board and the BFE chip. Table 4 shows the pin assignments of the connections required between the EK-RA4W1 board and the BFE device.

Table 4. Pin Assignments

Signal Name	BFE Pin	MCU Port	Evaluation Kit Pin
DGND	35, 36	--	J6 (-)
MISO/SDA	44	P100	27
MOSI	43	P101	26
SCL	42	P102	25
ADD/ /CS	41	P103	24
/ALERT	40	P111	17
/WAKEUP	39	P110	16
/RESET	38	P104	23
CMS0	34	P106	21

3.5 Terminal Emulator

The CLI of the Demo project enables the interaction of the user with the MCU to command the actions performed by the BFE. To access the CLI, the user requires serial communication between the PC and EK-RA4W1. Because the EK-RA4W is equipped with a USB-Serial converter IC, this communication can be handled as a COM port by a terminal emulator such as Tera Term. Table 5 shows the terminal setup for the project CLI.

Table 5. Settings of the Terminal

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

3.6 Use of Command-Line Interface (CLI)

The CLI is a text-based interactive access to command the execution of MCU routines that interact directly with the BFE. This section provides general guidelines on the use and features of the CLI. CLI Commands List shows the set of available commands and their functionalities.

When the EK-RAW4W1 is powered on, the terminal emulator program shows the CLI prompt **raa489206** indicating readiness to accept commands. Figure 7 shows the initial CLI prompt.

Figure 7. Initial CLI Prompt when MCU is Powered On

CLI commands have the following syntax:

```
[command-group] [sub-command] <value> <option> [LF or CR]
```

Command-group and **sub-command** are mandatory fields, whereas **value** and **option** (single character preceded by the hyphen minus) are optional parameters. Figure 8 shows some examples of command executions using the CLI. **Note:** Successful executions of commands produce the string [OK], whereas wrong or unsuccessful executions produce the string [ERROR] and its corresponding description.

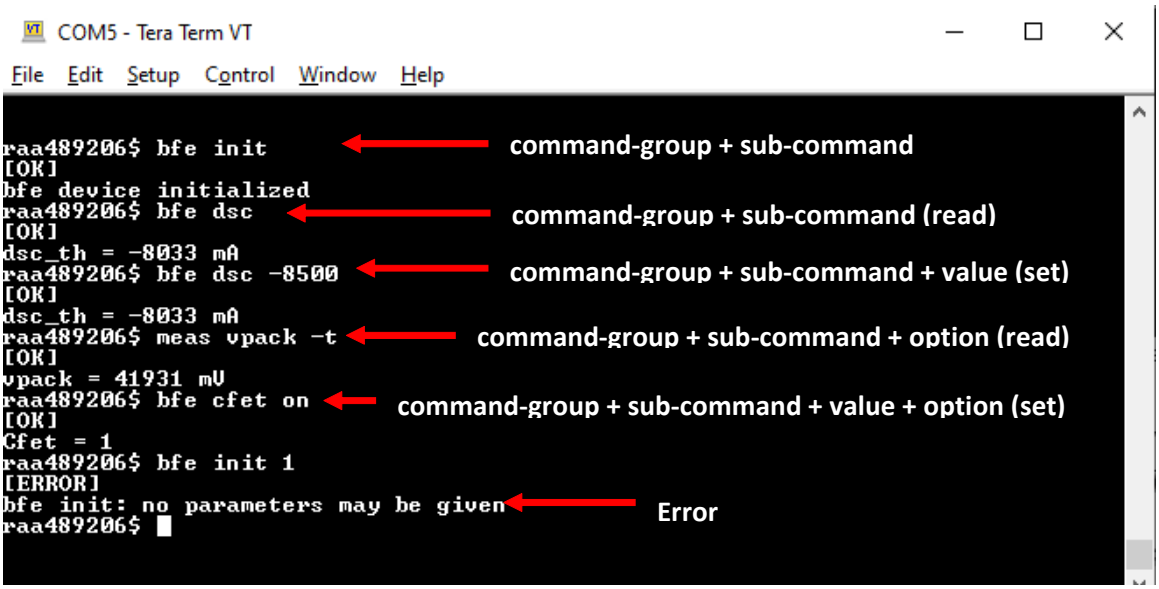

Figure 8. Examples of CLI Commands and their Syntax

The CLI includes Command-line completion. This feature enables the CLI to automatically fill partially typed commands. To use this feature, type the first few characters of a command, then press the **Tab** key. The CLI either completes the command or shows the commands that match the beginning of the typed characters. When the Tab key is pressed before typing any character, the CLI list all available commands or subcommands as a help feature. All commands include the implicit subcommand help, which displays a short description of the command use and the action it executes. Figure 9 depicts examples for the use of command completion and help features.

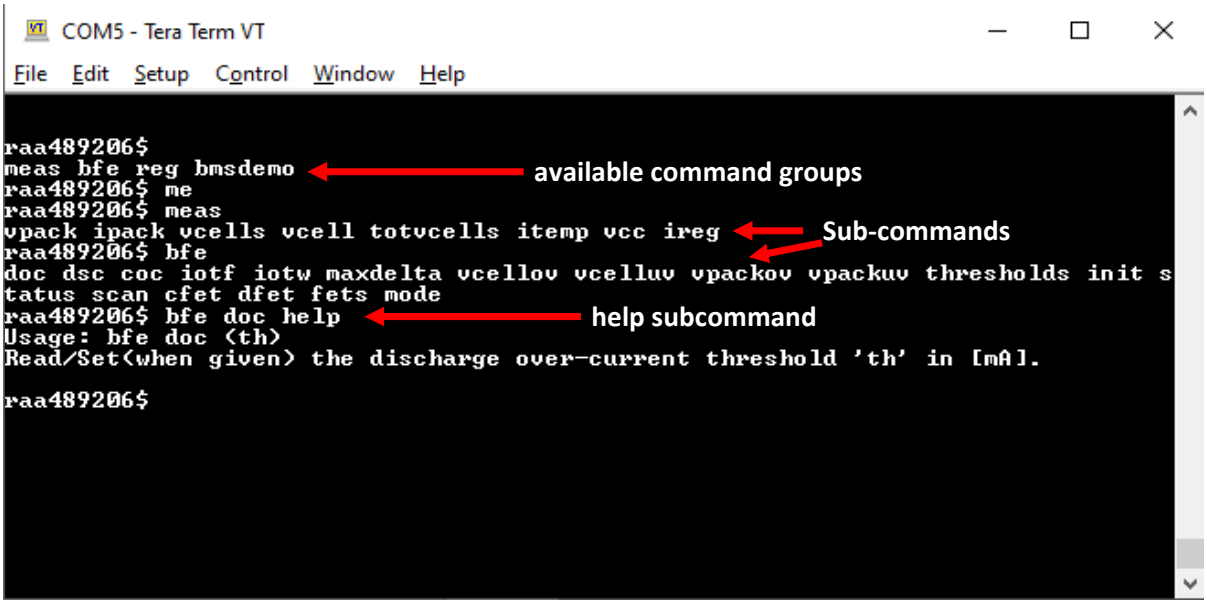

Figure 9. Command-Line Completion and Help Sub-command Features

4. CLI Commands List

This section lists the commands available in the CLI, specifically their parameters, and summarizes the actions on the BFE device. For detailed information on the commands, see the Application Note, *MCU Sample Code for Driving the RAA489206 16-Cell Battery Front End*. As convention, parameters within square brackets [ ] are mandatory, whereas parameters within angle brackets < > are optional.

Table 6. List of CLI Commands

Command	Sub-command	Parameters	Action on BFE	Examples
bfe	init[1]	none	Initializes device	bfe init
	doc	<none> <threshold>	Reads DOC threshold Sets DOC threshold (<0) in mA	bfe doc bfe doc -300
	coc	<none> <threshold>	Reads COC threshold Sets COC threshold (>0) in mA	bfe coc bfe coc 800
	dsc	<none> <threshold>	Reads DSC threshold Sets DSC threshold (< 0) in mA	bfe dsc bfe dsc -8000
	iotf	<none> <threshold>	Reads IOTF threshold Sets IOTF threshold in °C	bfe iotf bfe iotf 95
	iotw	<none> <threshold>	Reads IOTW threshold Sets IOTW threshold in °C	bfe iotw bfe iotw 85
	maxdelta	<none> <threshold>	Reads MAXDELTA threshold Sets MAXDELTA threshold in mV	bfe maxdelta bfe maxdelta 480
	vcellov	<none> <threshold>	Reads VCELLOV threshold Sets VCELLOV threshold in mV	bfe vcellov set vcellov 3000
	vcelluv	<none> <threshold>	Reads VCELLUV threshold Sets VCELLUV threshold in mV	bfe vcelluv bfe vcelluv 2000
	vpackov	<none> <threshold>	Reads VPACKOV threshold Sets VPACKOV threshold in mV	bfe vpackov bfe vpackov 40000
	vpackuv	<none> <threshold>	Reads VPACKUV threshold Sets VPACKUV threshold in mV	bfe vpackuv bfe vpackuv 30000
	status	<none> <-t> <clrfaults>	Reads status flags Triggers system scan before reading Clears all faults	bfe status bfe status -t bfe status clrfaults
	scan	<none> start stop delay delay <value>	Starts a complete system scan Starts continuous scan operation Stops continuous scan operation Reads the scan delay in ms Sets the scan delay to value = 64 | 128 | 256 | 512 | 1024 | 2049 | 4096 ms	bfe scan bfe scan start bfe scan stop bfe scan delay bfe scan delay 256
	thresholds	<none>	Reads and shows current device thresholds	bfe thresholds

Table 6. List of CLI Commands (Cont.)

Command	Sub-command	Parameters	Action on BFE	Examples
bfe (cont.)	cfet	<none> <1 | on | 0 | off> [value1] dfet [value2]	Reads CFET status Sets CFET status Sets CFET and DFET status	bfe cfet bfe cfet 1 bfe cfet on dfet off
	dfet	<none> <1 | on | 0 | off> [value1] dfet [value2]	Reads DFET status Sets DFET status Sets DFET and CFET status	bfe dfet bfe dfet 1 bfe dfet on cfet off
	fets	<none> <1 | on | 0 | off>	Reads D/C FETs status Sets both D/C FETs status	bfe fets bfe fets on
	mode	<none> <scan | idle | lpm | ship>	Reads BFE mode Sets BFE mode	bfe mode bfe mode scan
	cellcount	<none> <num>	Reads cells selected in the BFE Selects the given number of cells	bfe cellcount bfe cellcount 10
	rshunt	<none> <value>	Reads the shunt resistance in mΩ Sets the shunt resistance to value in mΩ	bfe rshunt bfe rshunt 30
reg	[reg-add]	<none> <hex-value> <-b> <all> <-s>	Reads the register with address reg-add Writes hex-value in the register reg-add Prints out the register value in binary format Prints out all register, sorted if -s is given	reg 0x01 reg 0x01 0x08 reg 0x01 -b reg all reg all -s
meas	vpack	<none> <-t>	Reads Vpack register in mV Triggers Vpack measure and waits for completion before reading Vpack value	meas vpack meas vpack -t
	ipack	<none> <-t>	Reads Ipack register in mA Triggers Ipack measure and waits for completion before reading Ipack value	meas ipack meas ipack -t
	vcells	<none> <-t>	Reads Vcells registers in mV Triggers Vcells measures and waits for completion before reading their values	meas vcells meas vcells -t
	vcell	[cell-number] [cell-number] <-t>	Reads Vcell in mV of the cell-number cell Triggers Vcells measures before reading Vcell	meas vcell 5 meas vcell 5 -t
	totvcells	<none> <-t>	Reads Vcells and return the total sum in mV Triggers Vcells measures and waits for completion before reading their values	meas totvcells meas totvcells -t
	itemp	<none> <-t>	Reads internal temperature in °C Triggers internal temperature measure and waits for completion before reading its value	meas itemp meas itemp -t
	vreg	<none> <-t>	Reads regulator voltage in mV Triggers regulator voltage measure and waits for completion before reading its value	meas vreg meas vreg -t
	ireg	<none> <-t>	Reads regulator current in mA Triggers regulator current measure and waits for completion before reading its value	meas ireg meas ireg -t
bmsdemo	--	---	Starts the sample Battery Management System	bmsdemo

Table 6. List of CLI Commands (Cont.)

Command	Sub-command	Parameters	Action on BFE	Examples
cb	none <enable> <disable>	- - -	Reads Cell Balancing Enable bit (CB EN) Enables cell balancing Disables cell balancing	cb cb enable cb disable
	veoc	<none> <value>	Reads the VEOC value in mV Sets VEOC to value in mV	cb veoc bfe veoc 4200
	ieoc	<none> <value>	Reads the IEOC value in mA Sets IEOC to value in mA	cb ieoc bfe ieoc 20
	auto	<none> <enable> <disable>	Reads the Automatic CB Enable bit Enables automatic CB Disables automatic CB	cb auto cb auto enable cb auto disable
	fets	<none> <ext> <int>	Reads CB FETs configuration Sets the use of external CB FETs Sets the use of internal CB FETs	cb fets cb fets ext cb fets int
	trigger	none	Triggers one cell balancing cycle	cb trigger
	mask	<none> <enable> <disable>	Reads the CB Masking bit (CB Mask) Enables mask feature Disables mask feature	cb mask cb mask enable cb mask disable
	eoc	<none> <enable> <disable>	Reads the CB end-of-charge bit (CB EOC) Enables CB EOC Disables CB EOC	cb eoc cb eoc enable cb eoc disable
	ieoc_en	<none> <enable> <disable>	Reads Current end-of-charge enable bit (IEOC EN) Enables IEOC EN Disables IEOC EN	cb ieoc_en cb ieoc_en enable cb ieoc_en disable
	chrg	<none> <enable> <disable>	Reads CB charge bit (CB CHRG) Enables CB CHRG Disables CB CHRG	cb chrg cb chrg enable cb chrg disable
	cell_state	<none> <hex-value>	Reads CB Cell State register Sets the CB Cell State register to hex-value (16 bits)	cb cell_state cb cell_state 0x0FF0
	mindelta	<none> <value>	Reads the CB Minimum Delta (CB Min Delta) Sets CB Min Delta to value mV	cb mindelta cb mindelta 50
	max_th	<none> <value>	Reads the CB Maximum (CBMAX) threshold Sets CBMAX to value mV	cb max_th cb max_th 4300
	min_th	<none> <value>	Reads the CB Minimum (CBMIN) threshold Sets CBMIN to value mV	cb min_th cb min_th 4300
	on_timer	<none> <value><s> <value><ms>	Reads CB On Timer (CBON) Sets CBON to value in seconds Sets CBON to value in milliseconds	cb on_timer cb on_timer 8 s cb on_timer 8 ms
	on_timer	<none> <value><s> <value><ms>	Reads CB Off Timer (CBOFF) Sets CBOFF to value in seconds Sets CBOFF to value in milliseconds	cb off_timer cb off_timer 8 s cb off_timer 8 ms

Table 6. List of CLI Commands (Cont.)

Command	Sub-command	Parameters	Action on BFE	Examples
soc	none <soci>	- <value>	Starts State-of-Charge (SOC) Application Starts SOC application using value in percentage (%) as initial SOC of the battery packs	soc soc 45.5

1. Execute the command **bfe init** first to initialize the BFE and set up the structures required for the other commands to work properly.

5. Revision History

Revision	Date	Description
1.01	Jan 31, 2023	Updated Tables 1, 3, and 6.
1.00	Aug 18, 2021	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 o any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer 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/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.