Description

The US134-118VBTPKEV1Z board demonstrates a proof of concept (POC) solution for a 20-28 cell battery management system (BMS). The board supports a wide input range of 40V-120V and output of 40V-118V with 20A of continuous discharge current< and can be increased to 60A with additional thermal management. Together with Renesas software resources, this board enables rapid prototyping and leads to shorter design cycles.

The RS-485 interface provides the flexibility to monitor system parameters like battery voltage, fuel gauge and load/charging current. It also accepts commands to control the system, monitor faults, and traceback errors during system shutdown.

The US134-118VBTPKEV1Z is an 8in x 5.5in 8-layer FR4 board with 2Oz copper on all layers. This board is compatible with the Renesas E2 emulator which provides a more comprehensive and efficient debugging environment. The E2 emulator grants flexibility to configure parameters like over-current threshold, cell balancing control, cell OV/UV limits, external temperature limits, faults masks, reduced cells operation, etc.

Kit Contents

- US134-118VBTPKEV1Z POC Board
- MCB-PS3-Z (x2) Battery Emulator Board

Features

- Supports 40V – 118V battery packs (20-28 cells).
- Wide charger input range 40V – 120V.
- 20A continuous discharge current (60 A with thermal management).
- Easily adaptable to operate with different battery cell counts.
- Features 2x daisy-chained RAA489204 to support high voltages up to 118V.
- ISL28025 DPM for monitoring system load/charge current.
- ISL89411 high speed dual channel MOSFET driver for controlling charge/discharge current.
- User input and diagnostics available via RS-485 (ISL32741E).
- RA2E1 low power MCU for providing overall system control.
- High voltage RAA223012 Buck and ISL80410 LDO for providing system power.
- Supports up to 8 thermistors for external temperature monitoring.
- Capable of autonomous charger/load detection and maintenance.
- Supports PMOD interface for optional Bluetooth connectivity.
- Protection features include Over-temperature (OT), Over-current (OC), Over -voltage (OV) and battery cell OV/UV.

System Boards

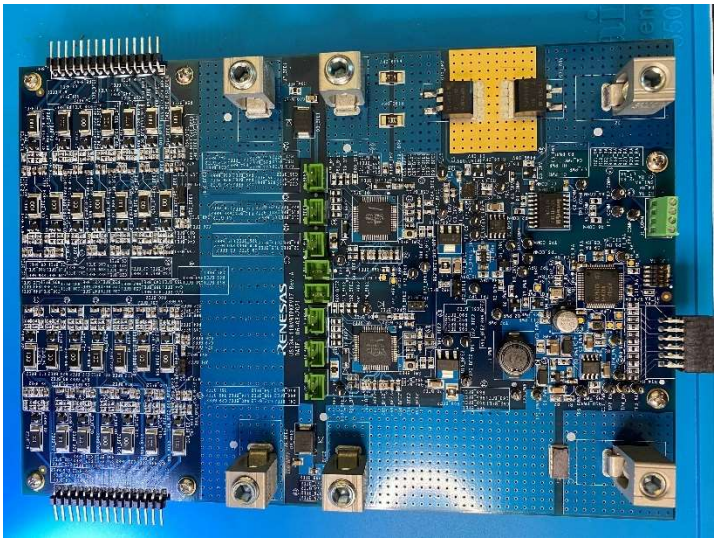


Figure 1. US134-118VBTPKEV1Z Board

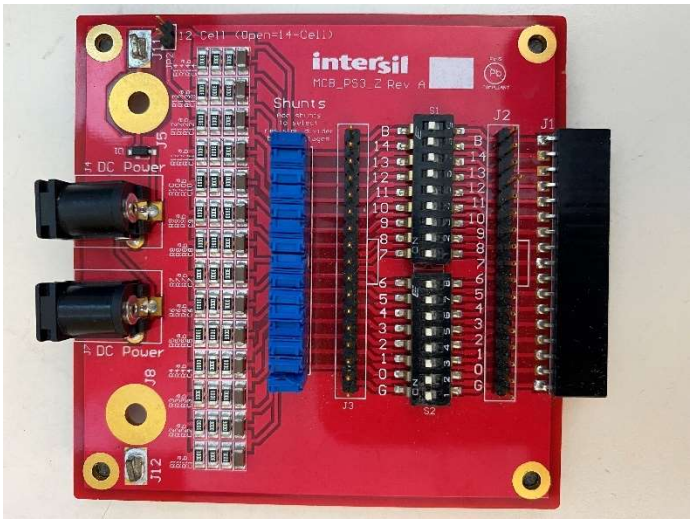


Figure 2. MCB-PS3-Z Battery Emulator Board

Contents

Reference Documents

Documentation which is referenced throughout this guide and other useful documents can be found below.

Document Name	Document No.
Getting Started Guide for e2 studio for RA	R20UT4891EJ0100
RA2E1 Group User's Manual: Hardware	R01UH0852EJ0110
Renesas Flexible Software Package (FSP) User's Manual	R11UM0137EU0081
ISL28025 Datasheet	FN8388
RAA223012 Datasheet	R16DS0139EU0200
ISL80410 Datasheet	FN8983
ISL89411 Datasheet	FN6798
ISL32741E Datasheet	FN8944

Table 1. Reference Documents

US134-118VBTPKEV1Z Overview

US134-118VBTPKEV1Z is intended to be a proof of concept (POC) solution for a battery management system, which can enable the designer to evaluate a 40V – 118V (20 – 28 cells) battery pack for charging and discharging applications, which in turn can lead to shorter design cycles and faster time to market. The US134-118VBTPKEV1Z incorporates multiple building blocks, along with software resources to provide a complete BMS solution, capable of standalone operation and requiring no external inputs for battery management and control.

The block diagram below highlights the main parts of the system:

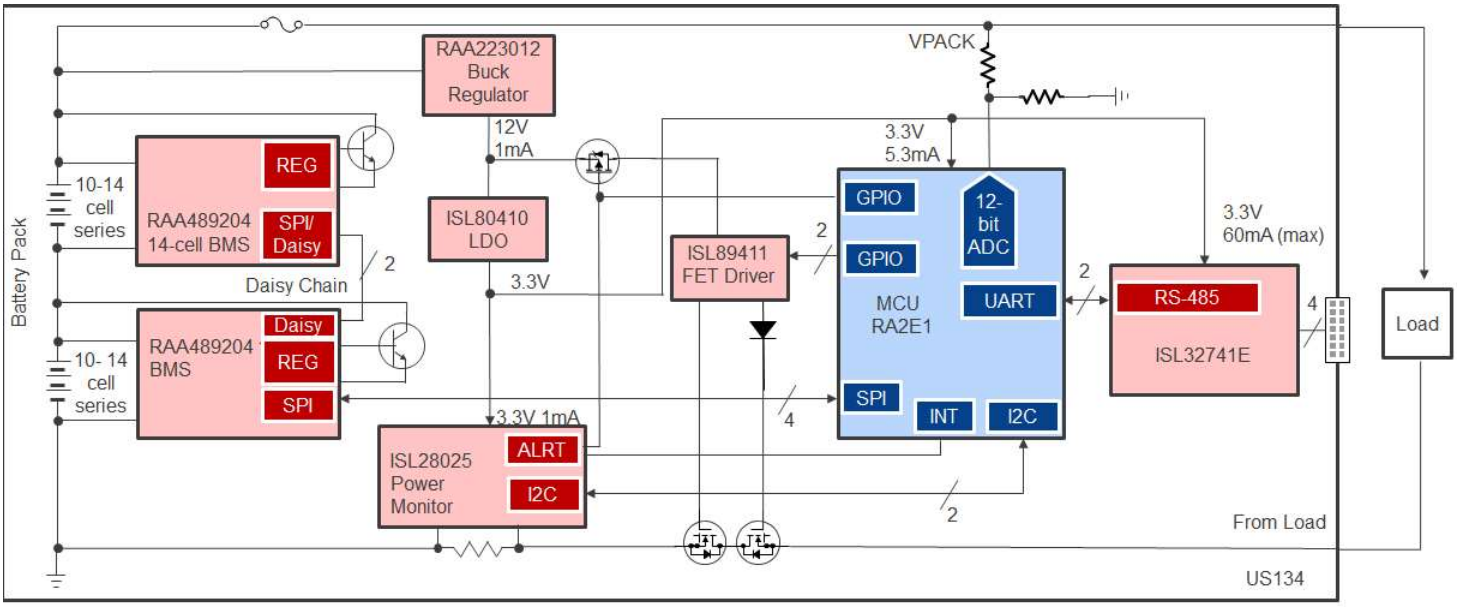


Figure 3. US134-118VBTPKEV1Z Block Diagram

The building blocks of the US134-118VBTPKEV1Z and their functionality are listed below:

- RAA489204 (x2) – Battery front end (BFE), which provides autonomous functions such as accurate cell voltage and temperature monitoring, cell balancing, and extensive system diagnostics. The devices are daisy chained so that the system can work with 20 – 28 cell battery pack(s).
- ISL28025 – Digital power monitor (DPM) is primarily used for current sensing during charging/discharging. It also provides voltage monitoring of the 12V rail generated by the RAA223012 and the 3.3V rail generated by the ISL80410. It provides fast-acting interrupts which can notify the MCU in case of an OC event and cut power to the ISL89411 FET driver, providing a fail-safe solution.
- RA2E1 – Renesas energy-efficient MCU which is the brains of the system. It contains the code which controls all the individual blocks of the US134-118VBTPKEV1Z system.
- RAA223012 – High-voltage buck regulator which generates the 12V for the system from the high voltage battery packs. It provides power to the ISL89411.
- ISL80410 – Low cost LDO which provides an accurate 3.3V for powering most of the system ICs.
- ISL89411 – Dual channel high speed MOSFET driver which is used for controlling the operation of the CFET and DFET.
- ISL32741E – RS-485 transceiver which is used for remote monitoring and system diagnostics.

The system also includes a P-Channel MOSFET for controlling the power to the ISL89411, which is turned OFF by the DPM (ISL28025) and the MCU during an Over-current (OC) event. The system features low side placement and control of the charging (CFET) and discharging (DFET) MOSFETs.

US134-118VBTPKEV1Z comes with software which is designed to effectively control the BMS solution. The code allows the US134-118VBTPKEV1Z board to operate autonomously, and detect the presence of a charger or load when plugged in. Multiple parameters like OC, OV, OT threshold levels, cell balancing controls, fault masking, reduced cell count operation etc. can be easily changed via accessing the header files in the code. An E2 emulator is required for programming any changes.

The US134-118VBTPKEV1Z can be used as a starting point to investigate target applications like home/solar backup systems, electric mobility solutions (hybrid electric vehicles, electric vehicles, electric motorcycles) and portable, battery-powered electronic equipment. The modular nature of the code can allow the user to experiment with different case studies.

Please note that the US134-118VBTPKEV1Z is NOT a turnkey solution to production. The solution does not guarantee all custom scenarios and users must exercise caution and are responsible for validating their final system.

Hardware Overview

The board has been designed to the following specifications:

- Input Charger voltage range = 40V – 120V
- Battery voltage range = 40V – 118V
- Battery cells = 20 – 28 cells
- Discharge current = 20A (60A with thermal management). Operation above 10A may require airflow.
- Charge current = Up to 10A

The board is designed to accommodate **two 14-cell** battery packs by default. These battery packs are connected in series when plugged into the POC board creating a high voltage system. The connectors for the battery cells, along with the connectors for the main power (VPACK1/2) are shown in Figure 4 below. BFE1 is referenced as the MASTER and BFE2 is referenced as the TOP device. BFE1 monitors the battery pack connected to VPACK1, along with the cell connections and BFE2 monitors the battery pack connected to VPACK2 (and the respective cell connections). These two BFEs are connected in a daisy chain configuration which allows battery monitoring and maintenance of the series-connected battery packs.

This board can also be configured to operate with less cells. Please refer to the “Operation with reduced battery cell count” section for details. The negative potential for VPACK1 is referenced as AGND in the schematics. This is the system GND for the entire board. All the ICs are referenced to this GND potential (except BFE2).

Main power for the board is derived from the battery packs. The RAA223012 buck section generates 12V from the series connected battery packs. The LDO further steps down the voltage to generate a regulated 3.3V (±1%) which powers all the ICs on the board, except BFE1 and BFE2 (BFE1/2 are powered by their own respective regulators). The 12V from the buck converter is used to power the ISL89411 FET driver which controls the CFET and DFET on/off operation.

All the other electrical blocks (IC’s) are highlighted in the image below. The RS485 (ISL32741E) secondary side is galvanically isolated from the rest of the system. The user needs to connect a power supply to power the secondary side of the RS485 transceiver. Please see “Auxi (RS-485)” for details.

Ensure that the correct polarity of the charger/load is followed when connecting either to the board. The board will automatically detect the connection of either a charger or a load and program the functionality accordingly. Please see “Charge detect, and Load detect Operation” for details. The maximum charging current for this board is limited to 10A. The maximum discharge current is limited to 20A. **Please note that based on ambient operating temperature, airflow may be required for operation at higher load current. It is highly recommended to use airflow when operating above 10A.** The max charging and discharging current limits can be changed by using Renesas debugging environment. Refer to “Debugging Mode” for details.

The connector for Renesas E2 emulator and PMOD is also highlighted below. If using the Renesas E2 emulator for debugging, connect the debugger first before connecting the batteries.

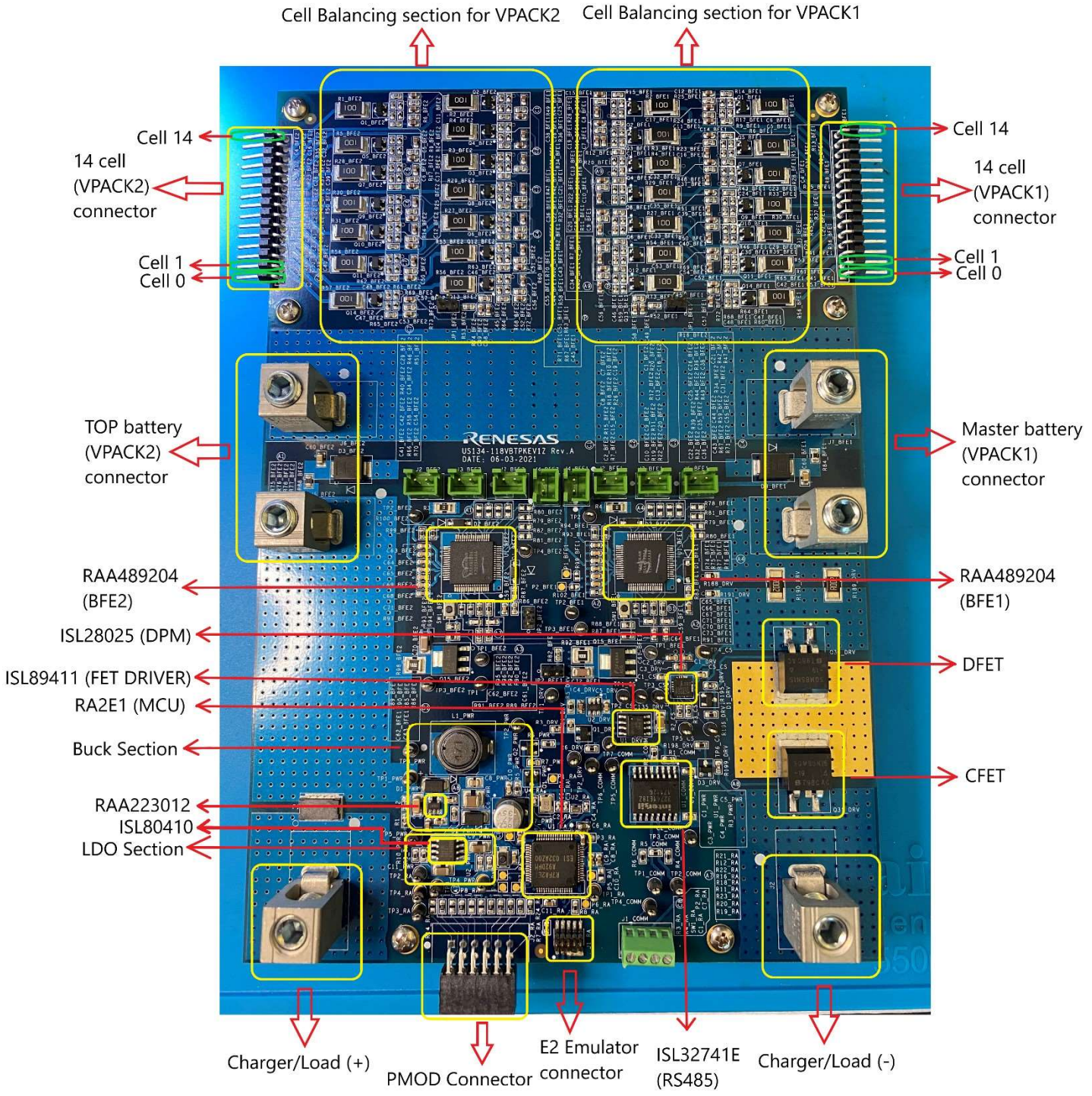


Figure 4. US134-118VBTPKEV1Z Board Overview

Exercise caution when operating the board with high voltage battery packs.

Recommended Equipment

- Charger. (DC Bench power supplies can be used in Constant power mode as alternative)
- DC power supplies (x3) with minimum 5V/1A sourcing capability
- DC electronic load capable of sinking up to 60A
- Renesas E2 Emulator
- Digital multimeters
- Current probe
- Oscilloscope
- Isolated USB to RS485 converter (Half duplex)
    - Serialcomm USB to Isolated RS485 / RS422 Adapter was used for this project

Quick Start Guide

The board is configured by default to work with two 14cells battery packs. For operation with reduced cell counts, please see “Operation with reduced battery cell count” on details to configure the board for reduced cell count. Eight thermistors have been populated on the board to allow the user to explore the functionality of the BFE.

The following terminology will be used throughout the document:

VPACK1 = Battery pack 1

VPACK2 = Battery pack 2

VPACK1+ = Positive terminal of Battery pack 1

VPACK2+ = Positive terminal of Battery pack 2

VPACK1- = Negative terminal of Battery pack 1

VPACK2- = Negative terminal of Battery pack 2

As mentioned earlier, the main GND for the system is AGND. All the ICs are referenced to this GND, except for U1_BFE2 which is referenced to AGND_TOP and the isolated side of U1_COMM which is referenced to GND_ISO. Make sure ESD straps are worn to prevent damaging any components on the board.

Operation with Battery Emulator

1. Renesas E2 emulator is required for operation with battery emulator. Firstly, check the emulator board with steps 2-7
2. Ensure all the shunts/jumpers are populated for JP1, J3 is OPEN and S1/S2 switches are all positioned to the left. See Figure 5 below.
3. Use two DC power supplies (PS1, PS2) to connect to the two respective battery emulators boards (MCB-PS3-Z) BEM1 and BEM2 as shown below.
4. Connect the positive terminal of power supply 1 (PS1) to J11 and the negative terminal of power supply 1 (PS1) to J12 of battery emulator board 1 (BEM1).
5. Make a similar connection between power supply 2 (PS2) and battery emulator board 2 (BEM2).
6. The battery emulator board consists of a resistor divider network connected between J11 and J12. The resistors are identical in value, so that the voltages between two successive pins on J2 are the same.
7. Adjust the voltage on PS1 and PS2 such that the voltage between consecutive pins is between 2V – 4.3V. Turn OFF PS1 and PS2 after verifying the voltage levels.

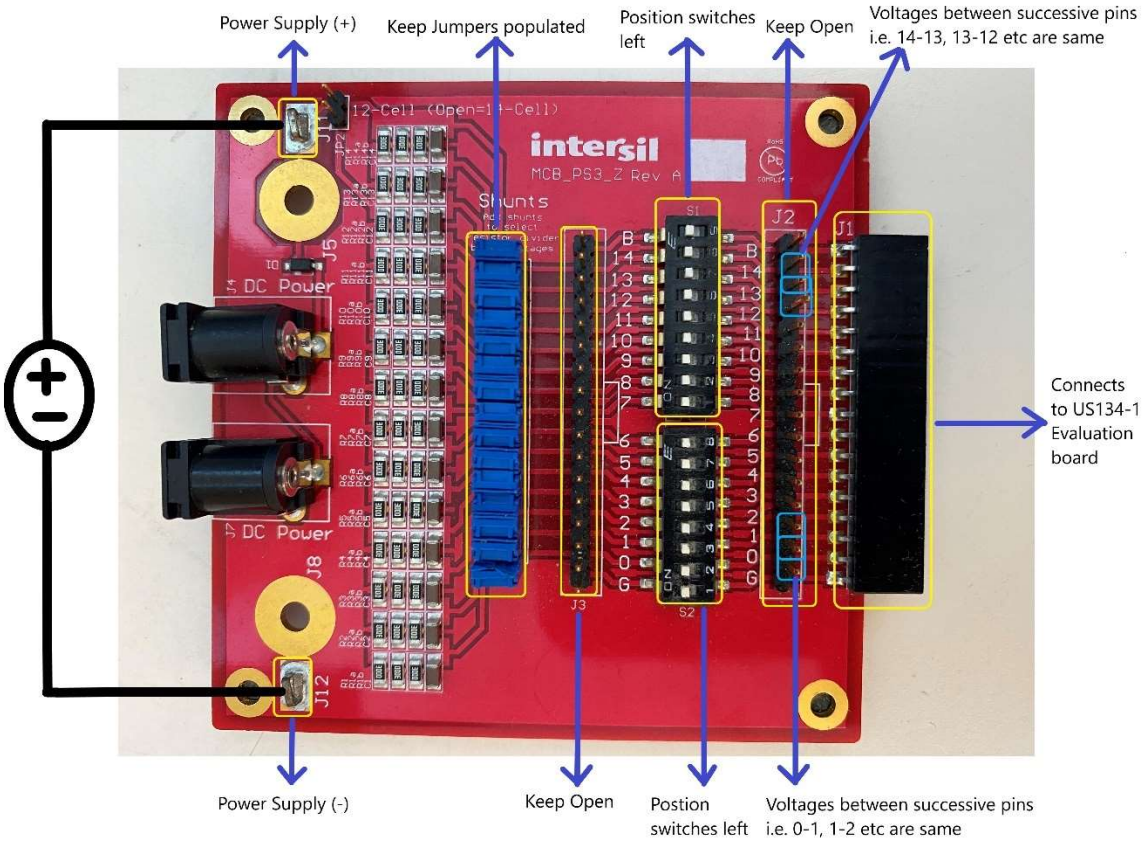


Figure 5. Battery Emulator Overview

8. When ready to connect, populate jumpers on JP1_BFE1, JP2_BFE1, JP1_BFE2 and JP2_BFE2 respectively.
9. Connect Renesas E2 emulator to the PC. Use a 20 pin to 10 pin ribbon cable.
10. Connect to J1_RA on the US134-118VBTPKEV1Z POC board. See Figure 4 for the debugger location.
11. Download the latest motherboard source code from www.renesas.com.

12. Connect battery emulator boards, BEM1 and BEM2 to J1_BFE1 and J1_BFE2 respectively as shown below. Make sure power supplies PS1, PS2 are OFF. **Note: BEM1 is flipped 180degrees, so PS1 will connect to J11 and J12 as shown in figure below.**

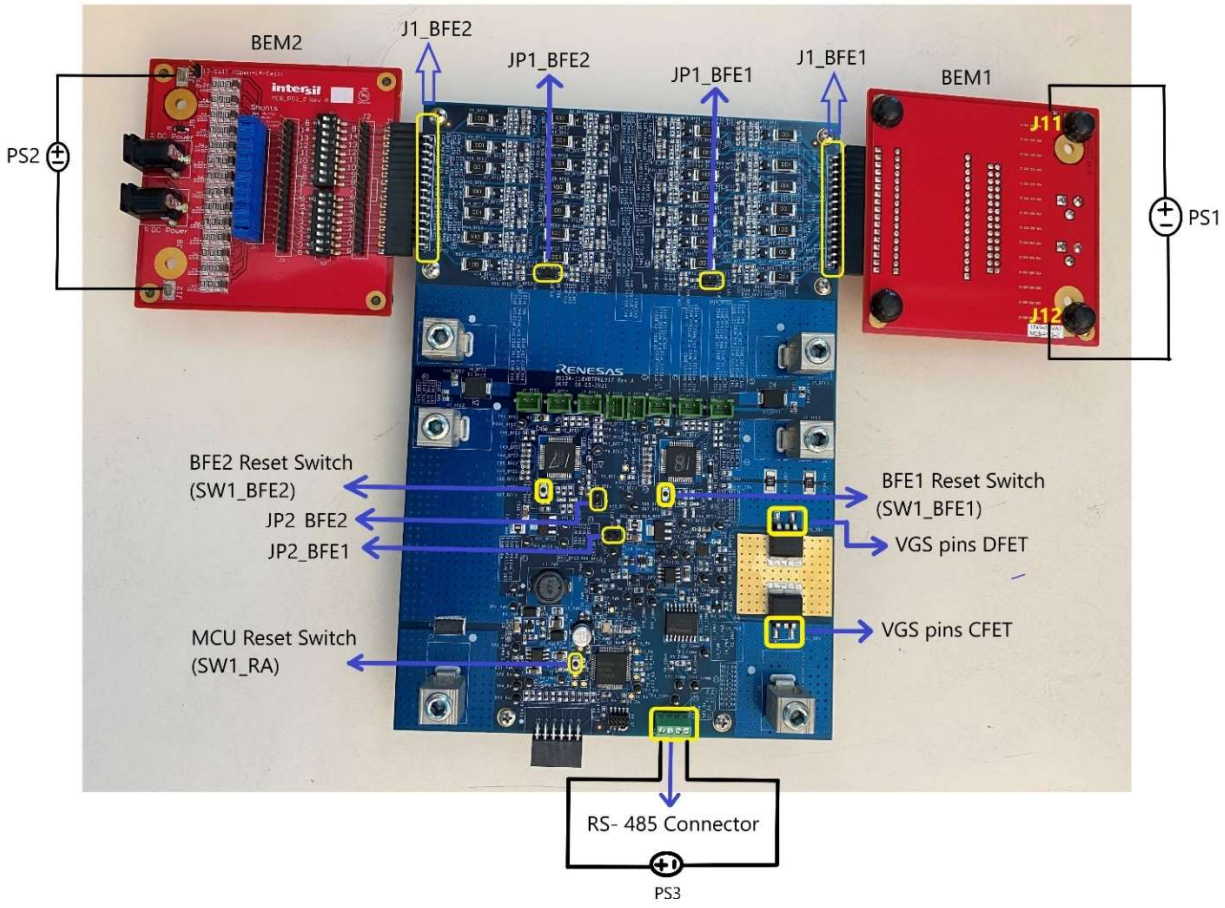


Figure 6. Battery Emulator Connection

13. Ensure pin 14 and pin 0 of the battery emulator board is connected to pin 15 and pin 1 of the US134-118VBTPKEV1Z POC board respectively. This means pins B and G on the battery emulator boards are floating.

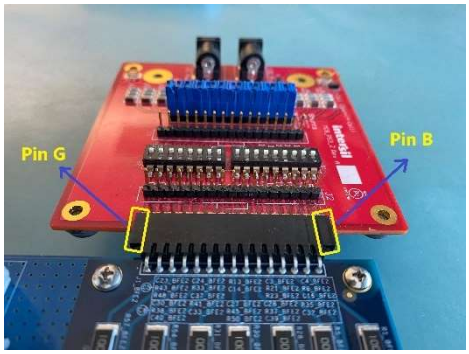


Figure 7. Battery Emulator Pin Connection

14. Connect the third power supply (PS3) to pins 1 and 4 of J1_COMM. Ensure that correct polarity is maintained and that the power supply is not enabled when making the connection.

15. Connect the differential lines of RS-485, “D+ or TX+/RX+” and “D- or TX-/RX- to pin 3 and pin 2 respectively of J1_COMM.
16. Open the RS-485 terminal window. See “Auxi (RS-485)” and “Standalone Mode” sections for details on RS-485 operation.
17. **Steps 14 – 16 are not required if RS-485 is not used.**
18. Turn ON PS1, followed by PS2.
19. Turn ON PS3 if using RS-485. Otherwise, skip this step.
20. **STOP**. Refer to “Demo Operation Guide” for details on how to run the E2 studio environment with battery emulators in standalone or with Renesas E2 emulator.

    Note that BFE1 and BFE2 enter sleep mode when system shutdown lasts for more than 60seconds. Toggle BFE RESET switches SW1_BFE2 and SW1_BFE1 to enable.
21. Use a voltmeter to probe TP2_PWR with respect to (w.r.t) TP5_PWR (AGND). The voltage should read 12V±5%.
22. Measure TP4_PWR w.r.t TP5_PWR (AGND). The voltage should read 3.3V± 1%.
23. Monitor TP3 w.r.t AGND and measure TP1 w.r.t AGND_TOP. Both voltages should read ~3.3V. This denotes that the BFE’s are operating normally and there are no faults to report.
24. Probe TP6_CS (SMBALERT2) and TP5_CS (SMBALERT1) w.r.t AGND. Both signals should read ~3.2V. This denotes that no OC faults are present.
25. Probe TP2_DRV (LD_DETECT) and TP1_DRV (CHRG_DET). They should read 3.3V and 0V respectively. Please see “Charge detect, and Load detect Operation” for details on operation of these circuits.
26. If using RS-485, observe Iout, fuel gauge, max/min voltages of BEM1 and BEM2, internal and external temperature measurements on the RS-485 terminal window. For more details on RS-485, please see “Auxi (RS-485)” and “Standalone Mode” for details.
27. Change the voltage on PS1 and PS2. Observe the change in the max/min cell voltages in the RS-485 console window. If the voltages between consecutive pins on either BEM1 or BEM2 reaches 4.16V, VEOC will be printed once in the RS485 console window.
28. If either of the voltages from BEM1 and BEM2 reaches the min/max UV limit or the sum of the voltage of PS1+ PS2 is less than 56V or greater than 121V, the system will shut down and the traceback error will be printed in the RS-485 console window. The BFE’s and the DPM will reset, and the system will go idle.
29. To restart the system, make sure BEM1/BEM2 voltage is between 2V – 4.3V.
30. If in standalone operation, toggle SW1_BFE2, SW1_BFE1 and SW1_RA, in that order.
31. If using Renesas E2 emulator, press disconnect on the E2 studio toolbar, toggle SW1_BFE2 and SW1_BFE1, and run the code again as described in “Debugging Mode”.
32. To disconnect the system, turn OFF PS2, followed by PS1. Remove BEM1 and BEM2 from the US134-118VBTPKEV1Z POC board. Turn OFF PS3.

Standalone Operation (with Battery Packs)

1. Remove the jumpers on JP1_BFE1, JP2_BFE1, JP1_BFE2 and JP2_BFE.
2. Connect a power supply to pins 1 and 4 of J1_COMM. Ensure that correct polarity is maintained and that the power supply is not enabled when making the connection.
3. Connect the differential lines of RS-485, “D+ or TX+/RX+” and “D- or TX-/RX- to pin 3 and pin 2 respectively of J1_COMM.
4. Open the RS-485 terminal window.
5. Make sure power supply (PS3) connected to pins 1 and 4 of J1_COMM is turned OFF.
6. Turn ON PS3 connected to the isolated side of RS-485. If not using RS-485 skip steps 3 - 5.
7. Connect VPACK1+ and VPACK1- to connectors J7_BFE1 and J6_BFE1 respectively.
8. Connect VPACK2+ and VPACK2- to connectors J5_BFE2 and J6_BFE2 respectively. Ensure correct polarity is maintained while making the connection.

9. Connect VPACK1 cell sense lines to J1_BFE1 and, then connect VPACK2 cell sense lines to J1_BFE2. Ensure the cells are correctly connected to the corresponding sense lines. See Figure 4.
10. Upon power up, the system first enters shut down. This is because the system detects an undervoltage lockout condition when the first battery is plugged in. For more details refer to RS-485 "Standalone Mode". A sample screenshot upon power up is given below:

Figure 8. System Shutdown Traceback

11. After connecting both the battery packs, toggle switches in the following order: SW1_BFE2, SW1_BFE1 and SW1_RA.
12. System will start cell balancing if the cells are not already balanced.

    Note: External thermistors are connected to the bottom of the POC board. If multiple cells are being balanced, the board can get hot and if the external temperature reaches 70degC, external temperature fault will trigger and shutdown the system. You can monitor the external temperature on the RS-485 console. Use airflow for cooling. Do not remove the thermistors.
13. RS-485 will continue to provide key system metrics at regular intervals.
14. Use oscilloscope probes to monitor TP1_DRV and TP2_DRV.
15. Connect an electronic load across the J1 and J2. Ensure correct polarity is maintained. See Figure 4 for details.
16. You should observe TP2_DRV (LD_DETECT) transition from high to low. This triggers the load detect interrupt and the system enters the discharge routine. CFET (Q31_DRV) and DFET (Q30_DRV) are turned ON and the batteries start sourcing current to the load.

    Note: After ~60 seconds, the system automatically shuts OFF the CFET and DFET if load is not turned ON or if the current is less than 100mA. Remove the load and plug it back in again to trigger the load detect interrupt. See "Charge detect, and Load detect Operation" for details.
17. Cell balancing stops during the discharge routine.
18. Turn ON the electronic load. Increase the current and monitor the status on the RS-485 console. **Note: It is highly recommended to use airflow when operating at high load current.**
19. Remove the load. The system automatically shuts OFF the CFET and DFET and cell balancing resumes after ~60 seconds.

    Note: Do not plug in the charger before the 60 second timer (user configurable – See the User Settings section) expires. Use a voltage probe and measure the gate to source voltage, VGS (pin 1 and 3) of CFET (Q31_DRV). Once the VGS drops below 1V, the charger is then ready to be plugged in. See figure 6 for location of CFET and DFET.
20. Connect the charger across J1 and J2. See Figure 4 to ensure correct polarity is maintained.

21. TP1_DRV(CHRG_DET) will transition from low to high. This triggers the charge detect interrupt and the system enters charging routine. CFET (Q31_DRV) and DFET (Q30_DRV) are turned ON and the charger starts to charge the batteries. **Note: It is recommended that the charger voltage be at least 3-5V higher than the sum of the series connected battery packs. Please see "Charge detect, and Load detect Operation" for details.**
22. Cell balancing continues during charging. RS-485 console will continue to output the health of the battery pack.
23. Disconnect the charger. After ~60 seconds, the system automatically shuts OFF the CFET and DFET. Cell balancing continues.
24. To disconnect the system, enter **"shutdown"** in the RS-485 terminal window. This will safely shut down the system and the system will go idle.
25. Remove VPACK2 cells sense lines, VPACK1 cells sense lines, VPACK2, and VPACK1, in that order.
26. If RS-485 is not used, remove VPACK2 cells sense lines. This will trigger an Open wire fault and safely shutdown the system.
27. Disconnect VPACK1 cells sense lines, VPACK2, and VPACK1, in that order. Turn OFF power to the RS-485 isolated side.

Operation with Battery Packs Using Renesas E2 Emulator

1. Remove the jumpers on JP1_BFE1, JP2_BFE1, JP1_BFE2 and JP2_BFE.
2. Connect Renesas E2 emulator to the PC. Use a 20 pin to 10 pin ribbon cable and connect to J1_RA on the US134-118VBTPKEV1Z POC board. See Figure 4 for the debugger location.
3. Download the latest motherboard source code from www.renesas.com.
4. Connect a power supply to pins 1 and 4 of J1_COMM. Ensure that correct polarity is maintained and that the power supply is not enabled when making the connection.
5. Connect the differential lines of RS-485, "D+ or TX+/RX+" and "D- or TX-/RX- to pin 3 and pin 2 respectively of J1_COMM.
6. Connect the RS-485. Open the RS-485 terminal window. If not using RS-485 skip steps 4 - 6.
7. Connect VPACK1+ and VPACK1- to connectors J7_BFE1 and J6_BFE1 respectively.
8. Connect VPACK2+ and VPACK2- to connectors J5_BFE2 and J6_BFE2 respectively. Ensure correct polarity is maintained while making the connection.
9. Connect VPACK1 cell sense lines to J1_BFE1 and, then connect VPACK2 cell sense lines to J1_BFE2. Ensure the cells are correctly connected to the corresponding sense lines. See Figure 4.
10. **STOP**. Refer to "Demo Operation Guide" for details on how to run the E2 studio environment with Renesas E2 emulator.

    Note that BFE1 and BFE2 enter sleep mode when system shutdown lasts for more than 60seconds. Toggle BFE RESET switches SW1_BFE2 and SW1_BFE1 and restart the debugger.
11. While in the main system loop, connect either a charger or load. Command prompts can be viewed in the console window in E2 Studio as well as in RS-485 terminal.
12. See steps 10 – 23 in "**Standalone Operation (With Battery Packs)"** for details on connecting a charger and load.
13. To disconnect the system, press the disconnect symbol in the toolbar. Toggle SW1_BFE2 and SW1_BFE1 if restart is required.
14. Remove VPACK2 cells, VPACK1 cells sense lines, VPACK2, and VPACK1, in that order.
15. To disconnect the system using RS-485, enter "shutdown" in the RS-485 terminal window. This will safely shut down the system and the system will go idle.
16. Remove VPACK2 cells sense lines, VPACK1 cells sense lines, VPACK2, and VPACK1, in that order.
17. Turn OFF power to the RS-485 isolated side.
18. Remove the debugger.

Operation with Reduced Battery Cell Count

US134-118VBTPKEV1Z POC board can be used with a 10 cell to 14 cell battery pack. Few modifications are required on hardware and in the code to allow the US134-118VBTPKEV1Z to operate with reduced battery cell count. Please note that the board only supports the same number of battery cells for VPACK1 and VPACK2. This means you cannot use dissimilar battery packs, for e.g. you cannot use VPACK1 with 12 cells and VPACK2 with 14 cells.

Use the following tables below to program the US134-118VBTPKEV1Z POC board to operate with less cells.

BFE1

No of cells	Populate 0Ω Resistor	Short J1_BFE1 pins
13	R74_BFE1, R75_BFE1	7 to 6
12	R76_BFE1, R77_BFE1 + 13 cell resistors	8 to 7 + 13cell connection
11	R78_BFE1, R79_BFE1 + 12 cell resistors	9 to 8 + 12cell connection
10	R80_BFE1, R81_BFE1 + 11 cell resistors	10 to 9 + 11cell connection

Table 2. BFE1 Resistors

BFE2

No of cells	Populate 0Ω Resistor	Short J1_BFE2 pins
13	R75_BFE2, R76_BFE2	7 to 6
12	R77_BFE2, R78_BFE2 + 13 cell resistors	8 to 7 + 13cell connection
11	R79_BFE2, R80_BFE2 + 12 cell resistors	9 to 8 + 12cell connection
10	R81_BFE2, R82_BFE2 + 11 cell resistors	10 to 9 + 11cell connection

Table 3. BFE2 Resistors

Code

No of cells	CELL_FAULT_MASK_USER (hex)	NUM_CELLS_USER	PIN_FAULT_MASK
13	0x0020	26	0x0080
12	0x0060	24	0x0080
11	0x00E0	22	0x0080
10	0x01E0	20	0x0080

Table 4. Code User Settings

In addition to cell and pin fault mask settings, overvoltage and undervoltage settings will be automatically adjusted by the NUM_CELLS_USER definition shown in Table 4. For example, absolute undervoltage is set to 2V and overvoltage is set to 4.32, which are multiplied by the number of cells set. These are system-level checks of the overall pack voltage which will be adjusted.

Populate zero-ohm resistors in the second column in Table 2, Table 3 and short J1_BFE2 and J1_BFE1 pins accordingly, for the corresponding battery cell count. This can be illustrated with an example below.

If VPACK1 and VPACK2 are **11 cell** batteries, the following resistors will be populated for BFE1:
R74_BFE1, R75_BFE1, R76_BFE1, R77_BFE1, R78_BFE1 and R79_BFE1.
Similarly, for BFE2, the following resistors will be populated:
R75_BFE2, R76_BFE2, R77_BFE2, R78_BFE2, R79_BFE2 and R80_BFE2.

Short pins 6, 7, 8 and 9 together on J1_BFE1, preferably on the bottom side of the board. Short the same pins together on J1_BFE2.

In the code, go to **bmsProfile.h** and make the following modifications to CELL_FAULT_MASK_USER to 0x00E0, according to Table 4.

The system can now accept two 11 cell battery packs.

```
system.c  bms.c  bms.h  startup.c  main.c  auxi.c  bmsProfile.h
39    * Macro definitions
41    #ifndef BMS_BMSPROFILE_H_
42    #define BMS_BMSPROFILE_H_
43
45    * Global Typedef definitions
47
48    // BMS register settings
49    #define OV_LIMIT_USER                (0x6E15)      // 4.3V, 4.19V = 0x6B44
50    #define UV_LIMIT_USER                (0x3333)      // 2V, 2.7V = 0x451E
51    #define EXTERNAL_TEMP_LIMIT_USER     (0x4CCC)
52    #define FAULT_SETUP_USER             (0x0FA8)      // Totalizer scan = 4, all external enabled
53    #define INTERNAL_TEMP_WARNING_USER   (0xBA80)
54    #define INTERNAL_TEMP_LIMIT_USER     (0xCE1B)
55    #define CELL_FAULT_MASK_USER         (0x0000)      // 0x0000 will not mask any faults
56
57    // cell balancing settings
58    #define CB_DELTA_MASTER              (0.003)       // cell balancing delta threshold for enabling balancing
59    #define CB_DELTA_TOP                 (0.01)       // cell balancing delta threshold for enabling balancing
60    #define CB_DISABLE_WAIT_USER         (30)          // time to allow after disable/enable CB in milliseconds
61    #define CB_ENABLE_WAIT_USER          (100)         // time to allow after disable/enable CB in milliseconds
62
63    // battery limit settings
64    #define VPACK_OV_LIMIT_USER          (117.6)
65    #define VPACK_UV_LIMIT_USER          (75.6)
66    #define OV_LOCKOUT_USER              (121)         // all 28 cells over 4.5V
67    #define UV_LOCKOUT_USER              (56)          // all 28 cells under 2V
68    #define OV_ABS_USER                  (121)         // any num of cells
69    #define UV_ABS_USER                  (40)          // any num of cells
```

Figure 9. Cell Fault Mask Code

If **10 cell** battery packs are being used, populate all the resistors (zero-ohm) for 11 cells as stated above, plus R80_BFE1, R81_BFE1 for BFE1 and R81_BFE2, R82_BFE2 for BFE2 respectively. Change CELL_FAULT_MASK_USER to 0x01E0.

If **13 cell** battery packs are being used, populate zero-ohm for only R74_BFE1, R75_BFE1 (BFE1) and R75_BFE2, R76_BFE2 (BFE2). CELL_FAULT_MASK_USER needs to be programmed to 0x0020.

Charge Detect and Load Detect Operation

The system detects the presence of a charger or load using interrupts which are triggered by hardware circuitry on board. The image below shows the series connected battery packs along with the charge/load detection circuitry. CHRGE/DSCHRGE– is connected to PACK1- (or AGND) via R3_DRV, R4_DRV and R5_DRV when the CFET (Q31_DRV) and DFET (Q30_DRV) are OFF.

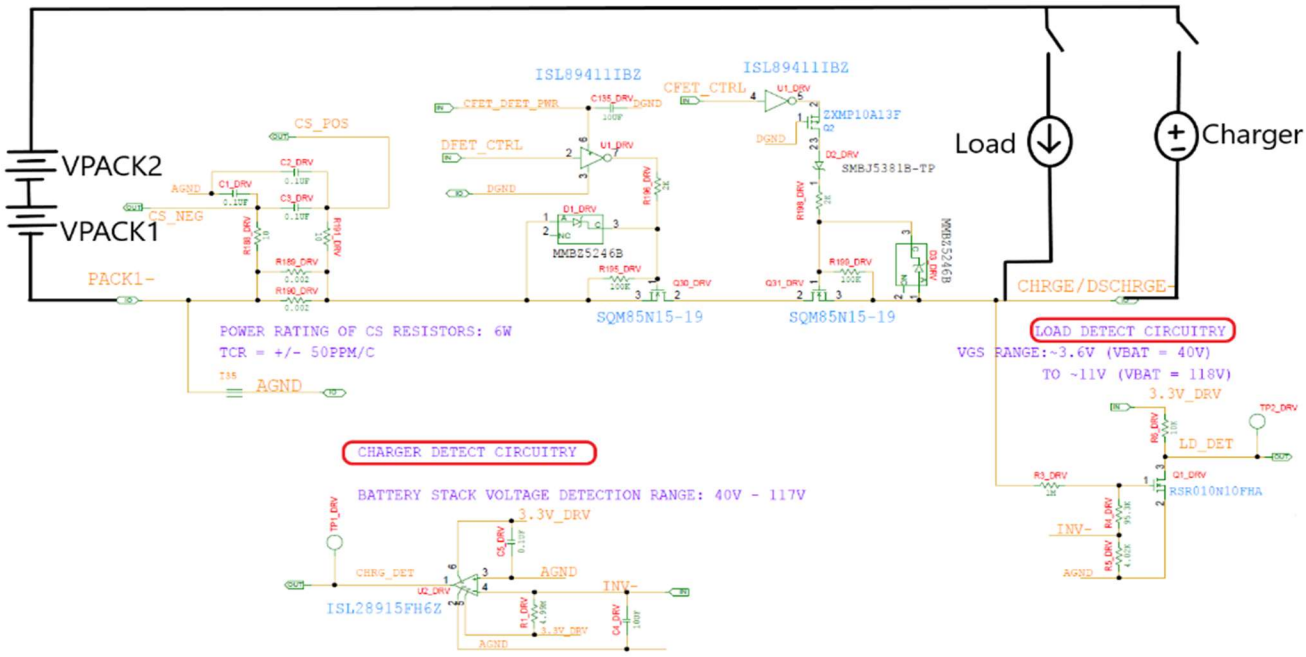


Figure 10. Charge and Load Detect

Load Detection Circuitry

LD_DET connects to pin 12 (P411) of the MCU. By default, when no load (or charger) is connected, the pin is pulled to 3.3V via 10kΩ resistor R6_DRV.

When a load is connected to the connected across J1 and J2, CHRGE/DSCHRGE- voltage is pulled to "VPACK2 + VPACK1". Since the MOSFETs are OFF, the current flows through the series connected resistors (R3_DRV, R4_DRV and R5_DRV) to AGND. This creates a voltage drop across VGS of Q1_DRV, which results in the FET being turned ON. This pulls LD_DET signal low and the MCU registers the interrupt and puts the system in a discharge routine.

The load detection circuitry can detect loads up to 5kΩ.

Charge Detection Circuitry

CHRG_DET connects to pin 13 (P410) of the MCU. When no charger (or load) is connected, the pin is pulled low. CHRGE/DSCHRGE- is floating and hence, INV- is pulled very close to AGND. R1_DRV is added to provide some noise margin (offset from AGND) in case CHRGE/DSCHRGE- picks up some noise during charger or load plug in.

When a charger is plugged in across J1 and J2, the current flows via R5_DRV, R4_DRV and R3_DRV to CHRGE/DSCHRGE- since the CFET and DFET are turned OFF. This creates a voltage drop across R5_DRV which in turn trips the comparator and CHRG_DET goes high (or 3.3V). The MCU detects the interrupt and puts the system in the charging routine.

There is a minimum differential voltage required between the charger and the battery packs (VPACK1 + VPACK2) to trigger the charge detection circuitry. It is recommended that the charger voltage should be at least 3-5 volts higher than the sum of the series connected battery packs.

Please note that a good test setup is very important when testing the POC board. Use short and low gauge wires to connect to J1 and J2. Try and minimize sources of noise on the same power strip.

Overcurrent Protection

US134-118VBTPKEV1Z POC board features a robust and redundant over current protection circuitry. The digital power monitor (U1_CS) is responsible for monitoring the charge/discharge current. There are three protection mechanism inherent in the system to safeguard against over-current protection:

1. SMBALERT1
2. SMBALERT2
3. Threshold detection within the code

SMBALERT1

SMBALERT1 pin on the DPM (U1_CS) is an open drain pin which requires an external pullup. During normal operation SMBALERT1 is pulled high to 3.3V via R7_PWR. NPN Q2_PWR is operating in the non-saturation region with ~30uA collector current. This creates a voltage drop across R5_PWR, which turns ON Q1_PWR, providing power to the FET driver U1_DRV.

During an OC event, SMBALERT1 is pulled low, which pulls the base of Q2_PWR to 0. This shuts of the NPN and the voltage across VGS of Q1_PWR discharges slowly according to the RC time constant of C10_PWR and R5_PWR.

This cuts off power to the FET driver, which eventually turns OFF the CFET and DFET. The RC time constant is set such that there is ~400ms delay before the VGS of Q1_PWR starts to decrease. This delay is inserted to prevent any inrush current to falsely trigger the OC event and mitigate any potential noise issues which might occur while connecting the load/charger. See Figure 11 for details.

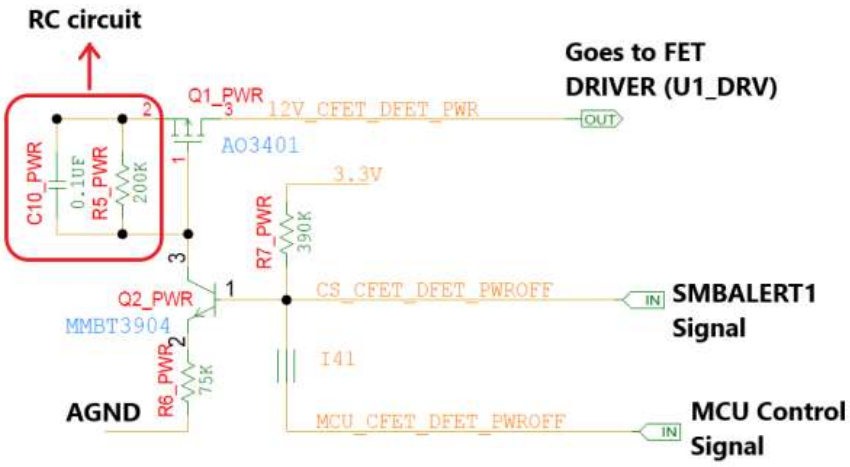


Figure 11. SMBALERT Circuit

SMBALERT2

SMBALERT2 pin of the DPM is a push pull output which goes to pin 60 (P004) of the MCU. This signal is used as an interrupt for the MCU to detect an OC event. The default state of the pin during normal operation is high (3.3V). During an OC event, the pin is pulled low, which triggers an interrupt.

This interrupt is serviced in the "**sys_oc_handle**" which initially checks for inrush current by incorporating a blanking time of 50ms. If the fault condition persists, the system enters "**sys_shudown().**" In **"sys_shutdown()"** the MCU turns OFF the CFET and DFET control signals to the FET driver and pulls the MCU_CFET_DFET_PWROFF signal low.

Pulling MCU_CFET_DFET_PWROFF signal low has the same effect as SMBALERT1 signal pulling low, which shuts OFF the NPN and turns OFF Q1_PWR after an RC delay.

Please note that when Q1_PWR is turned OFF, and the voltage starts falling below ~11.5V, the ISL28025 can detect a under voltage condition and shutdown the system.

These hardware and software redundancies guarantee a failsafe operation.

```
 * @brief       Checks for over-current (with inrush current handling) using the current reading from the DPM
void sys_oc_handle(void)
{
    g_dpm.iout = dpm_read_iout();

    // check for OC flag set by dpm
    if (imon_oc_fault)
    {
        // blanking time for inrush OC
        imon_oc_fault = 0;
        R_BSP_SoftwareDelay(OC_BLANKING_USER, BSP_DELAY_UNITS_MILLISECONDS);
        g_dpm.iout = dpm_read_iout();

        // check iout reading again for OC threshold
        if ((0 == sys_iout_dir) && (g_dpm.iout > OC_THRESHOLD_USER))
        {
            g_dpm_fault.Bit.OCF = 1;
            func_flag = 7;
            fault_flag = 1;
            sys_traceback(func_flag, fault_flag);
            sys_shutdown();
        }
```

Figure 12. OC Handle Code

Threshold Detection Within the Code

In "**dpm_update**" during an OC event, the code compares the present value of the current with the OC_THRESHOLD_USER value that is set by the user. This check, if violated will set the "g_dpm_fault.Bit.OCF to 1", which in turn will be processed by "sys_fault_handle()" and lead to system shutdown.

See the "System Faults Overview" and "Main System Flow" sections for details.

The above-mentioned OC faults are processed only for the **discharge current** when the system is connected to the load. For protection against an OC event during **charging**, the code allows the user to set the over current charging limit "SYS_IOUT_OC" in dpmProfile.h.

If the user exceeds the charging current limit, the "g_dpm_fault.Bit.OCF" is set to 1 and the system enters the fault handler and shuts down the system.

```
 * @brief       Updates the DPM readings and checks for DPM register faults and overcurrent faults
void dpm_update(void)
{
    dpm_clear_faults();

    //R_BSP_SoftwareDelay(100, BSP_DELAY_UNITS_MILLISECONDS);

    // take register readings
    g_dpm.vshunt = dpm_read_vshunt();
    g_dpm.iout = dpm_read_iout();
    g_dpm.voutAux = dpm_read_vout_aux();
    g_dpm.temp = dpm_read_temp();
    g_dpm.vout = dpm_read_vout();

    // update the faults by reading the status registers
    g_dpm.dpmFault.Word = dpm_read_status_word();
    g_dpm.voutStatus.Byte = dpm_read_status_vout();

    // system checks iout reading for OC threshold (Charging)
    if ((1 == sys_iout_dir) && (g_dpm.iout < (-1 * SYS_IOUT_OC)))
    {
        func_flag = 3;
        g_dpm_fault.Bit.OCF = 1;
    }
```

Figure 13. DPM Update Code

The default programmed OC values for the board are:

Discharge = 21.25A

Charging = 10A

Software Overview

The following sections will give an overview of the software implementation of the US134-1, which is based on the Renesas RA Family's Flexible Software Package (FSP). These sections will detail the demo project's code structure, the system's software modules, the main system flow, and provide an operation guide for the demo. Additional information regarding the demo such as the software API, pin functions, and fault handling will also be provided.

Demo Project Code Structure

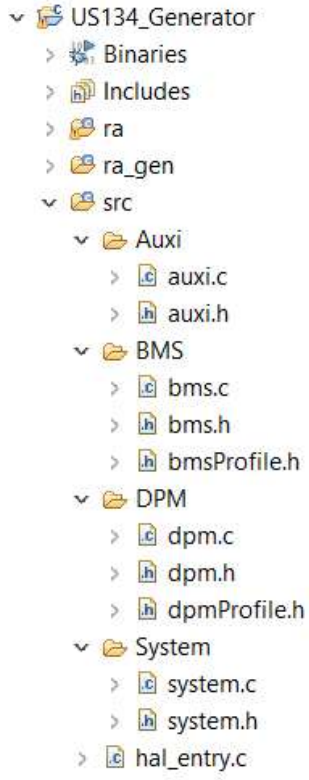
The US134-118VBTPKEV1Z demo project is designed to be a highly modular solution, where each device has its own associated module folder that can be easily configured independently of the other modules or ported to other end applications.

The project is split into 4 main modules:

- Auxi – includes the UART communications driver which interfaces with the ISL32741E RS-485 transceiver
- BMS – RAA489204 device driver code for battery management which includes the SPI communications driver
- DPM – ISL28025 device driver code for power monitoring which includes the I2C communications driver
- System – main system code that enables all other drivers (ADC, interrupts, timers, etc.) and implements system flow

Each module folder contains the C source files and header files for that module. The DPM and BMS modules also have an additional Profile header file for specific user configurations. Refer to the "User Settings" section for more details regarding user configurations.

Figure 14 below shows the structure of the project in e2 studio.



- ra – includes automatically generated files for FSP drivers
- ra_gen – includes main file and generated files for FSP driver settings
- src – contains the module code which utilizes FSP drivers
    - Auxi – RS-485 driver code
    - BMS – battery management system code
    - DPM – digital power monitor code
    - System – main system code
- hal_entry.c – start of code execution which calls system main

Figure 14. US134-118VBTPKEV1Z Code Structure

Figure 15 below shows the general code structure in terms of its dependencies. Execution begins in hal_entry.c, which calls the main function in system.c. From here, the main system flow begins, which in turn uses the Auxi, BMS, and DPM modules to execute the demo. All associated header files reference the lower-level Flexible Software Package (FSP) drivers.

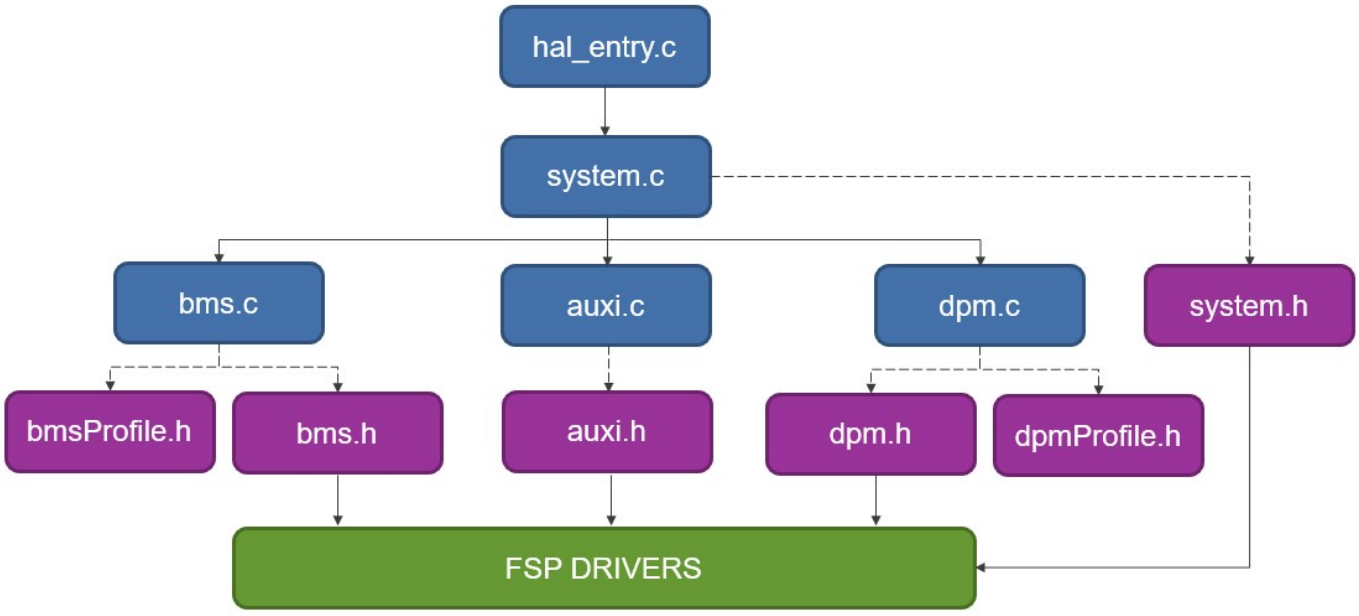


Figure 15. Code Dependency Graph

System modularity is achieved by routing all calls to the Auxi, BMS, and DPM modules through the System module, such that these modules do not make calls to one another and therefore have no interdependency.

Software Module Overview

In this section a description of the contents of each software module is provided, including their features and project scope.

System

The System module contains the main system algorithm which is described in the “Main System Flow” section. This module is responsible for initializing and setting up all the drivers which are used in the main algorithm, such as the system timers, the analog-to-digital converter, and the system interrupts. This module also makes calls to the other modules to initialize and set up themselves. After initialization, this module is responsible for monitoring the system, handling faults, and shutting down the system in the event of faults. When a charger or a load is plugged in to the board, this module will be responsible for detecting and enabling the connection, as well as monitoring the status of charge/discharge.

The US134-118VBTPKEV1Z demo project is capable of communicating with the user via RS-485 from a host PC as well, and this module handles pushing out diagnostics and interpreting user commands. Refer to the “Demo Operation” section for more information.

Auxi (RS-485)

The Auxi module is responsible for initializing the UART driver and reading from/writing to the RS-485 transceiver. When the system pushes out information to the user, this module takes the output string and utilizes the FSP’s UART driver to write the data to the RS-485. When data is received from the RS-485 transceiver, each byte is read and placed into an input buffer until the ‘Enter’ key is pressed. The UART buffer contents are then copied over to a system buffer and the UART buffer is cleared to prepare for the next transmission. Interpretation of the received data is handled by the System module.

When using the RS-485 transceiver with the demo project, the following features are available:

- Printing system diagnostics such as fuel gauge and temperature readings
- Printing out the contents of DPM and BMS fault registers via user command
- Forcing the system to shutdown via user command
- Printing out the system error traceback when the system goes into shutdown

BMS (Battery Management System)

The BMS module is a comprehensive device driver capable of establishing a daisy chain of 2 devices and handling reading from/writing to the registers of these devices. This module is responsible for initializing the FSP SPI driver and setting up each BMS device with the user-configured settings. All transmissions to and from the BMS devices require a Cyclic Redundancy Check (CRC), and this module can compute and verify CRC-16 and CRC-32 packets.

After setup, the module provides the following features on each device:

- Performing a variety of scans (voltages, temperatures, open wires, etc.)
- Performing various device commands (sleep, wakeup, reset, etc.)
- Reading cell and pack voltages
- Reading internal and external temperatures (up to 4 external temperatures)
- Reading device faults (overvoltage, undervoltage, open wires, etc.)
- Performing cell balancing (finds minimum cell voltage, compares to threshold delta to select cells to balance)
- Reading from and writing to all device registers with a single or multi-register read/write

During the main system flow, each BMS device is fully scanned, readings are taken, and faults are checked. Cell balancing is then performed if a load is not connected and battery emulators are not being used.

DPM (Digital Power Monitor)

The DPM module is a device driver that can monitor the current and voltage in the system for added protection. This module is responsible for initializing the FSP I2C driver and setting up the DPM device with the user-configured settings.

After setup, the module provides the following features:

- Performing various device commands (clear faults, reset, etc.)
- Reading the 12V system voltage
- Reading the 3.3V VCC voltage
- Reading the system current
- Reading the device's internal temperature
- Reading device faults (overvoltage, undervoltage, overcurrent, etc.)
- Reading from and writing to all device registers

During main system flow, the DPM device readings are taken, and faults are checked. When a load or charger is connected, the device is updated to change the overcurrent threshold current direction accordingly.

Main System Flow

This section describes the main system algorithm which is implemented in the System module. The algorithm is responsible for initializing and setting up each device module and FSP driver in the system, before entering the main loop where the system is monitored to safely charge or discharge the battery. A flowchart is provided in this section which outlines the algorithm at a high level, and then each step is described in further detail.

Algorithm Flowchart

Figure 16 below describes the algorithm at a high level.

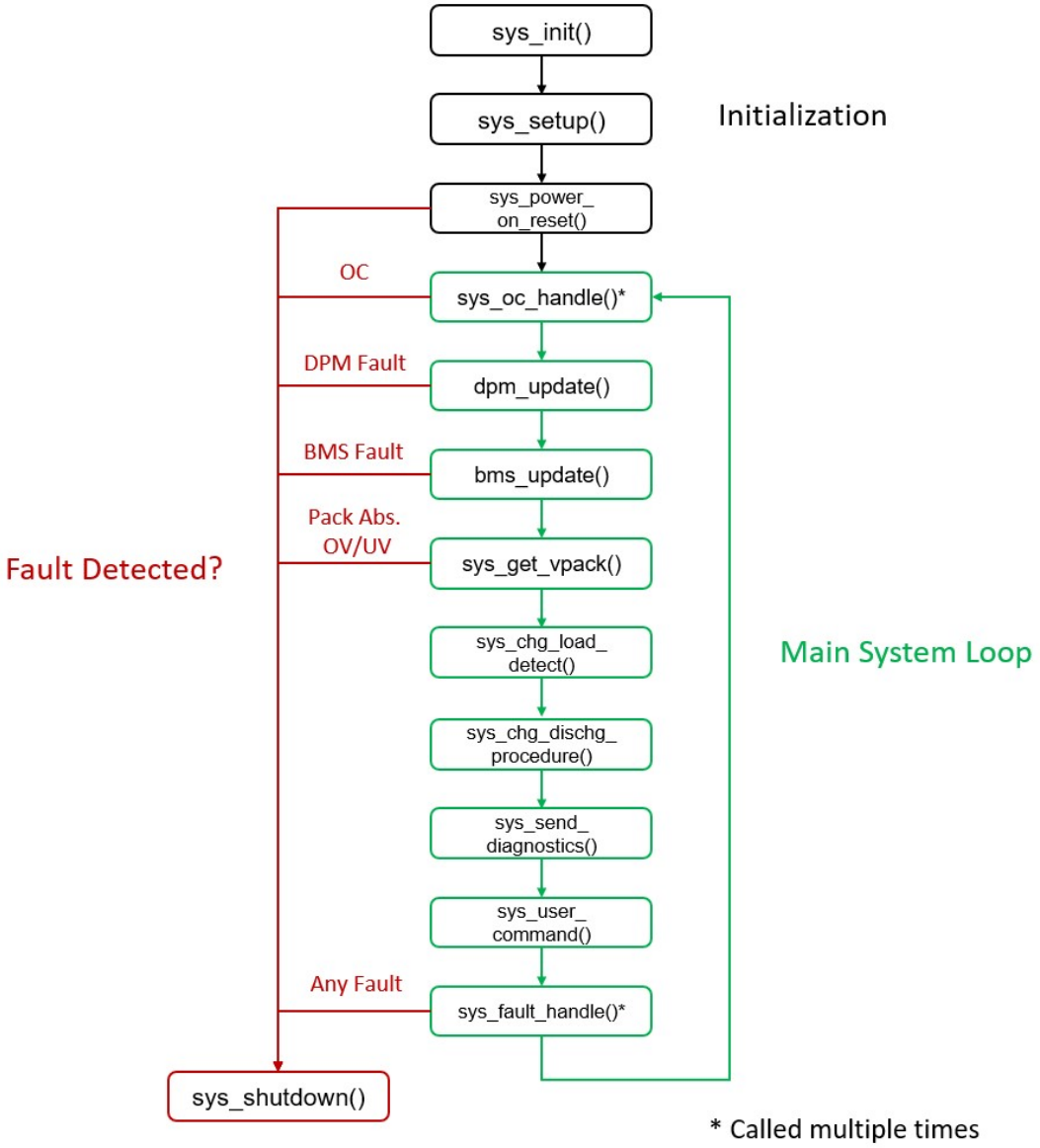


Figure 16. System Algorithm Flowchart

The functions outlined in Figure 16 are described below.

- Sys_init()
  - Turn off CFET/DFET and turn FET driver on
  - Initialize all FSP drivers, and the 3 modules' communication drivers (UART, I2C and SPI)
- Sys_setup()
  - Set up DPM and BMS with default register settings and test communications
  - Reset interrupt flags (fault pins, charger/load detect)
  - Start system timer (for timing RS-485 transmissions and monitoring current)
- Sys_power_on_reset()
  - Take DPM readings and check for 12V and 3.3V signals
  - Read all cell voltages
  - Call sys_get_vpack to check battery levels
- Sys_oc_handle()
  - Read DPM current and check for OC fault flag
  - Check for fault misfire caused by inrush current in both current directions
  - Shutdown if OC fault occurred correctly
- Dpm_update()
  - Clear faults and take readings (system voltages, current, temperature)
  - Check for faults
  - Check for OC manually
- Bms_update()
  - Clear faults
  - Scan devices and take readings (cell/pack voltages, internal/external temperatures)
  - Check for faults
  - Perform cell balancing algorithm if no faults detected
    - Check if VEOC and IEOC (voltage/current end-of-charge) were reached and stop balancing when IEOC limit hits
    - If not, find minimum cell voltage
    - Compare each cell to a user-defined threshold voltage delta to select cells to balance
    - Start balancing for selected cells
- Sys_get_vpack()
  - Read pack voltage measured by BMS and System ADC
  - Set UV/OV limits if reached
  - Compare pack voltage values and shutdown if differential exceeds user-defined threshold
  - Shutdown if pack voltage exceeds absolute pack voltage limit thresholds

- Sys_chg_load_detect()
  - Check if load/charger connection was detected
  - Check if the other was not already connected and the UV/OV limit was not hit
  - If not, set current threshold direction and turn on CFET/DFET to begin charging/discharging
  - Call dpm_update() and bms_update() to update DPM and BMS
  - Check if load/charger was connected and current is present
- Sys_chg_dischg_procedure()
  - When discharging:
    - Check if load was removed by checking if current is near zero
    - If current dropped to near-zero or never came up, begin counting up
    - If count threshold is reached, turn off FETs and reset flags and timers
    - Check if current comes back before threshold is hit to reset counter
    - If OV was hit and CFET was turned off, start counting positive current readings
    - If positive reading count threshold is reached or current exceeds FET body diode, turn CFET back on
  - When charging:
    - Check if charger was removed by checking if current is near zero
    - If current dropped to near-zero or never came up, begin counting up
    - If count threshold is reached, turn off FETs and reset flags and timers
    - Check if current comes back before threshold is hit to reset counter
- Sys_send_diagnostics()
  - Check if 15 seconds (user configured) passed and push out current and max/min cell voltages on RS-485
  - Check if 25 seconds (user configured) passed and push out internal/external temperatures on RS-485
  - Check if 100 seconds (user configured) passed and push out fuel gauge for battery packs on RS-485
  - Check if VEOC/IEOC were set and push out notification on RS-485
- Sys_user_command()
  - Check if data was received from RS-485
  - Compares the input data string with a known list of commands
  - If the string matches with a command, execute the command
    - ‘shutdown’ will shut down the system
    - ‘faults’ will print the DPM and BMS faults
    - ‘help’ will print these commands
- Sys_fault_handle()
  - Check if any faults occurred in the system, DPM, or BMS
  - Shutdown the system if any faults occurred (except for open wire faults when using battery emulators)

- Sys_shutdown()
    - Turn off CFET, DFET, and system driver FET
    - Initiate a hard reset in both BMS devices, and a soft reset in the DPM devices
    - De-initialize all system and communications drivers
    - Reset all flags, interrupts, and faults
    - Go idle

The main system loop begins after sys_init(), sys_setup(), and sys_power_on_reset() have concluded, and loops back after the sys_fault_handle() has determined there are no faults in the system or from any modules. If any faults occur, they will lead to a system shutdown from either the function where the fault occurred, or from sys_fault_handle(). Refer to the "System Faults Overview" section for details on the possible faults. Refer to the "Software API" section for more details on the demo project functions.

Demo Operation Guide

This section provides a guide to operating the demo from the perspective of the software. For details on the hardware connections and board bring up procedure, refer to the "Quick Start Guide" section. This section assumes that you have completed this procedure and connected the battery packs and cell balancing connectors. For debugging mode, the E2 emulator must be used, and the RS-485 transceiver can also be optionally used. For standalone mode, the RS-485 transceiver must be connected to a 5V power supply.

Debugging Mode

After connecting the battery packs, the cell balancing connectors, and the E2 emulator for debugging mode, the hardware connections are complete. The RS-485 transceiver can also be optionally used. The steps below outline the process of running the demo project in debugging mode. Refer to the Getting Started Guide for e2 studio for RA in the "Reference Documents" section for additional guidance on using e2 studio.

- Install e2 studio for RA with an FSP version of 2.3.0 or later
- Import the US134-118VBTPKEV1Z project into the e2 studio workspace
- Enable printf statements in the e2 studio Renesas Debug Virtual Console (Figure 17)
    - Right-click on the project > Properties > select 'Settings' under 'C/C++ Build'
    - Select the Tool Settings tab > select 'Preprocessor' under 'GNU ARM Cross C Compiler'
    - Under 'Defined symbols', select Add > type 'DEBUG_MODE' > OK > Apply and Close
        - Alternatively, edit the 'DEBUG_MODE0' symbol to 'DEBUG_MODE'
    - To disable printf statements for standalone mode, change or the remove the 'DEBUG_MODE' symbol

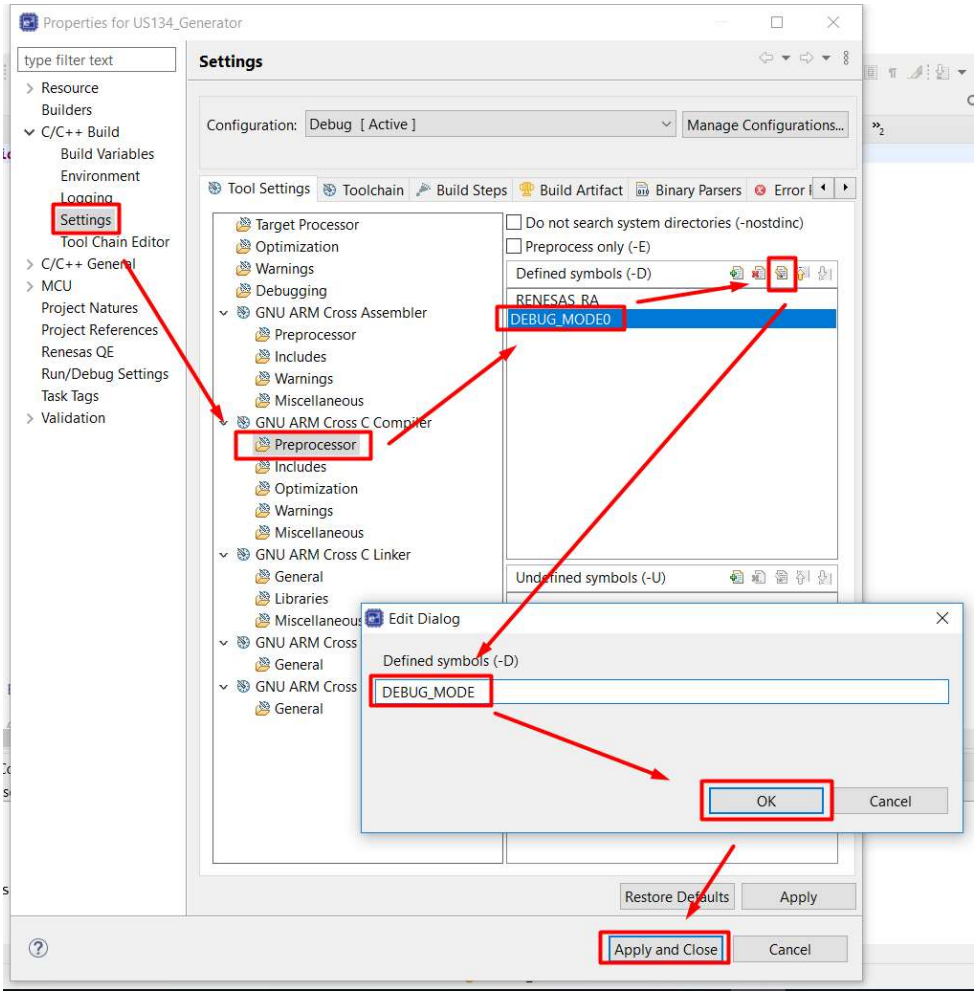


Figure 17. Debug Mode Setting

- **If using a battery emulator**, set the 'bat_emulator' variable to 1 on line 56 of system.c
  - Ensure that the 'bat_emulator' variable is set to 0 when using battery packs
  - Cell balancing will not occur
- **If using a battery emulator**, set the FAULT_PIN_MASK on lines 138 and 139 of bms.c to '0x0080'
  - This will mask the open wire fault caused by the emulators such that the fault pin doesn't assert
- Adjust any user settings as required such as for different numbers of cells
  - Refer to the "User Settings" section for more details
- Build the project
- Open the Renesas Debug Virtual Console in the Debug perspective
  - Select the Debug perspective in the top-right corner of the e2 studio window
  - Select the Renesas Views tab > Debug > Renesas Debug Virtual Console

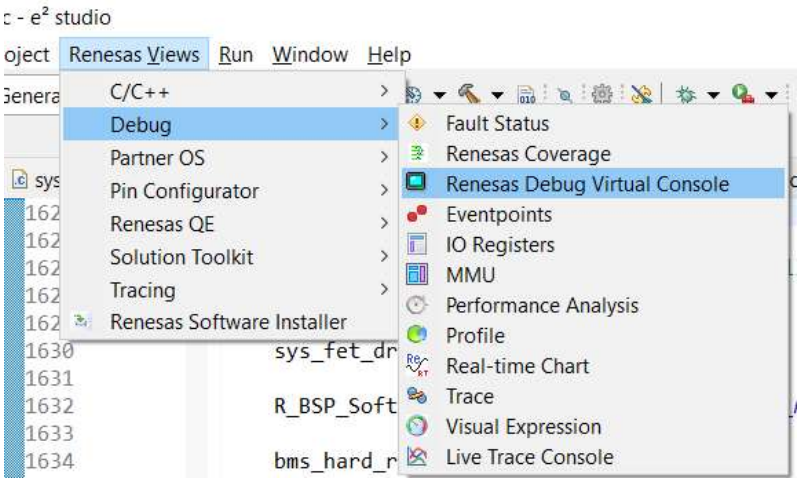


Figure 18. Renesas Debug Virtual Console

- Debug the project
- (Optional) Open a serial emulator such as Tera Term and connect to the RS-485 transceiver
    - Refer to the “Standalone Mode” section below for setting up the RS-485 communications
- When the debugger has finished flashing the code into the board, select Resume (F8) twice to begin operation
- The code will print ‘DPM setup complete’ and ‘BMS setup complete’ and then transition into the main loop
    - This will be printed in the virtual console and the serial emulator
    - If the code does not print ‘BMS setup complete’, reset the BMS devices with the reset switches
        - The BMS devices will go to sleep after 60 seconds of inactivity
        - The Watchdog Timer is set to max (1520 minutes) in bms_setup()
    - Reset the code with the Restart button and select Resume (F8)
- During the main loop, the user can interact with the system
    - Diagnostics will be printed in the serial emulator periodically
    - The user can type commands into the serial emulator
        - ‘shutdown’ will shut down the system
        - ‘faults’ will print the DPM and BMS faults
        - ‘help’ will print these commands
        - If commands don’t match, a message will be printed to the serial emulator
    - Plug in a load or charger and observe the charge/discharge procedure
        - Detection messages are printed to the virtual console
        - After connecting it, the user should turn on the load/charger within the timeout period
            - See the User Settings section for more details
        - After disconnecting the load/charger, wait until the 60-second timeout period to reset the system to idle
            - A message will be printed to the virtual console/serial emulator to signal removal

    - Place breakpoints in the code to view system values in e2 studio
        - Right-click variables such as ‘g_bms_master’ and select ‘Add Watch Expression’
        - View their values in the Expressions tab
        - Right-click the expression and select ‘Enable Real-time Refresh’ to update values in real time
- If any fault is encountered, the system will shutdown
    - Shutdown traceback logs are printed to the virtual console/serial emulator
    - Refer to the “System Faults Overview” section for more information
- To restart the system, select Terminate in e2 studio to disconnect and select Debug again
    - Alternatively, select the Restart button

Standalone Mode

After connecting the battery packs, the cell balancing connectors, and the RS-485 transceiver with 5V power supply, the hardware connections are complete. The steps below outline the process of running the demo project in standalone mode. If switching from debugging mode to standalone mode, refer to Figure 17 on how to disable debugger print statements.

Note that the code will not operate if debugging mode print statements are included during standalone mode.

- Open a serial emulator such as Tera Term and connect to the RS-485 transceiver
    - The serial settings are as follows (Figure 19):
        - Baud rate: 115200
        - Data packet: 8N1 (8 data bits, no parity, 1 stop bit)
    - The terminal settings are as follows (Figure 20):
        - Receive: CR
        - Transmit: CR + LF
        - (Optional) Local echo enabled

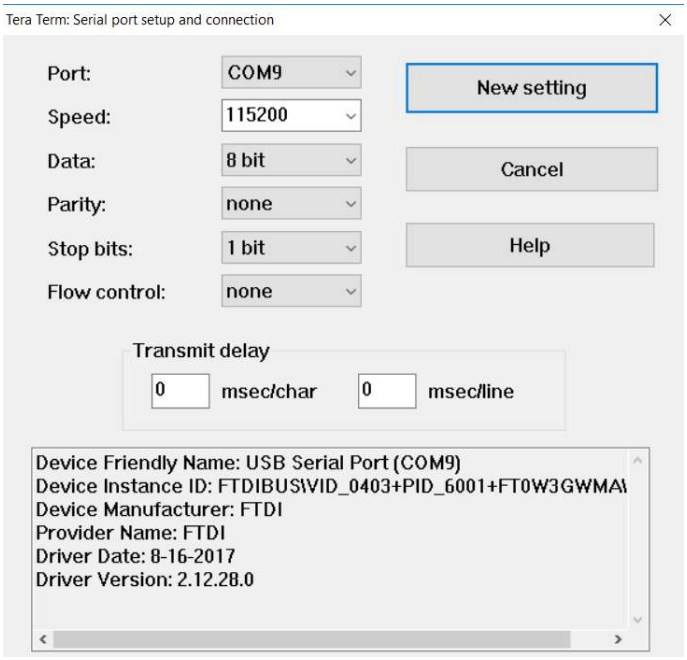


Figure 19. Serial port Settings

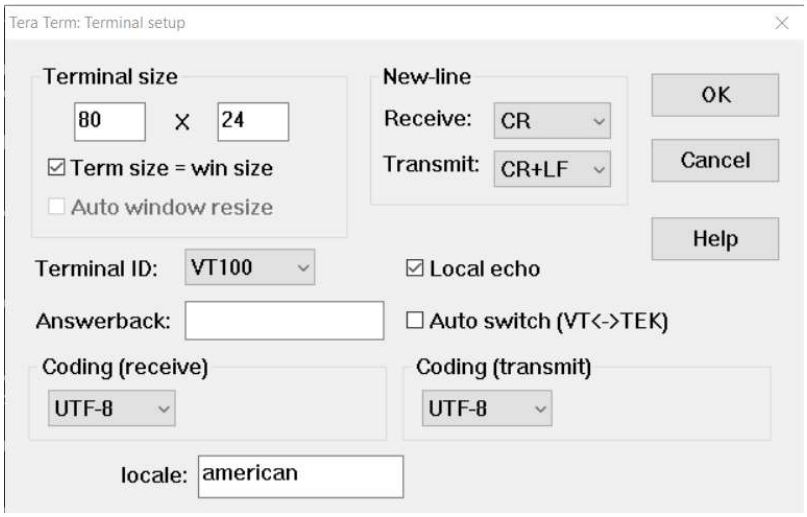


Figure 20. Terminal Settings

- Reset the BMS devices and then the system MCU with the reset switches
- The code will print 'DPM setup complete' and 'BMS setup complete' and then transition into the main loop (Figure 21)
    - This will be printed in the virtual console and the serial emulator
    - If the code does not print 'BMS setup complete', reset the BMS devices with the reset switches
        - The BMS devices will go to sleep after 60 seconds of inactivity
        - The Watchdog Timer is set to max (1520 minutes) in bms_setup()

Figure 21. Serial Emulator Printout

- During the main loop, the user can interact with the system
  - Diagnostics will be printed in the serial emulator periodically (Figure 22)
  - The user can type commands into the serial emulator
    - 'shutdown' will shut down the system
    - 'faults' will print the DPM and BMS faults
    - 'help' will print these commands
    - If commands don't match, a message will be printed to the serial emulator
  - Plug in a load or charger and observe the charge/discharge procedure
    - Detection messages are printed to the virtual console
    - After connecting it, the user should turn on the load/charger within the timeout period
      - See the User Settings section for more details
    - After disconnecting the load/charger, wait until the 60-second timeout period to reset the system to idle
      - A message will be printed to the virtual console/serial emulator to signal removal

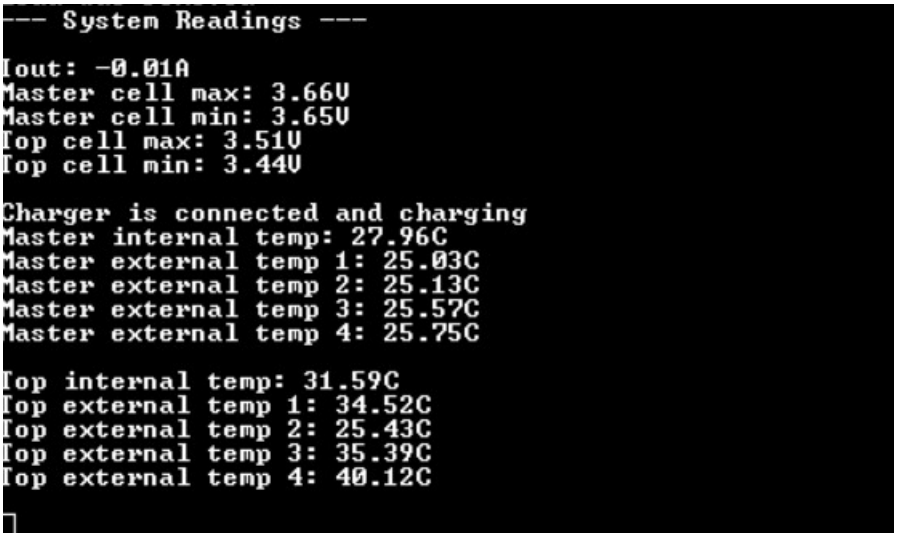


Figure 22. System Readings

- If any fault is encountered, the system will shut down

- Shutdown traceback logs are printed to the serial emulator (Figure 23)
- Refer to the “System Faults Overview” section for more information

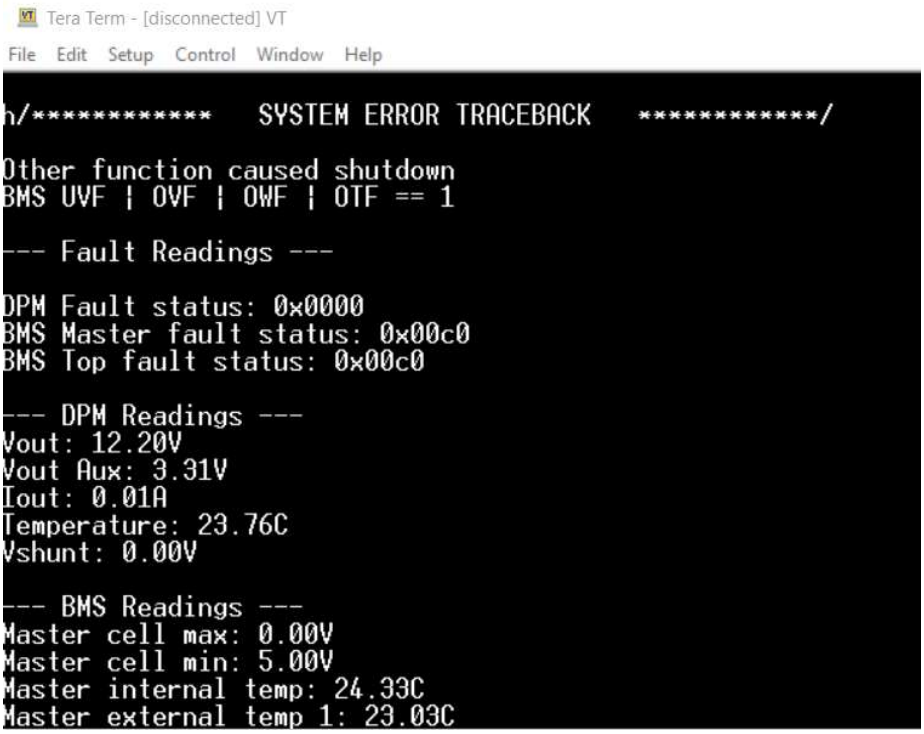


Figure 23. Shutdown Traceback

- To restart the system, reset the BMS devices and then the system MCU with the reset switches

User Settings

This section outlines all the configurable user settings in the demo project. Configurable settings are mainly included in the Profile header files for the DPM and BMS, with several exceptions. These configurations are defined macros with the ‘_USER’ suffix. The list of user configurable settings for the DPM and BMS devices is not exhaustive; the most commonly used settings are included in the Profile headers, but all register settings in the DPM and BMS setup functions can also be adjusted directly. Refer to the datasheets for these devices in the “Reference Documents” section for guidance on register settings and their values. Note that the user will require an E2 emulator to make changes to the demo project.

Figure 24 below shows the Profile header file contents.

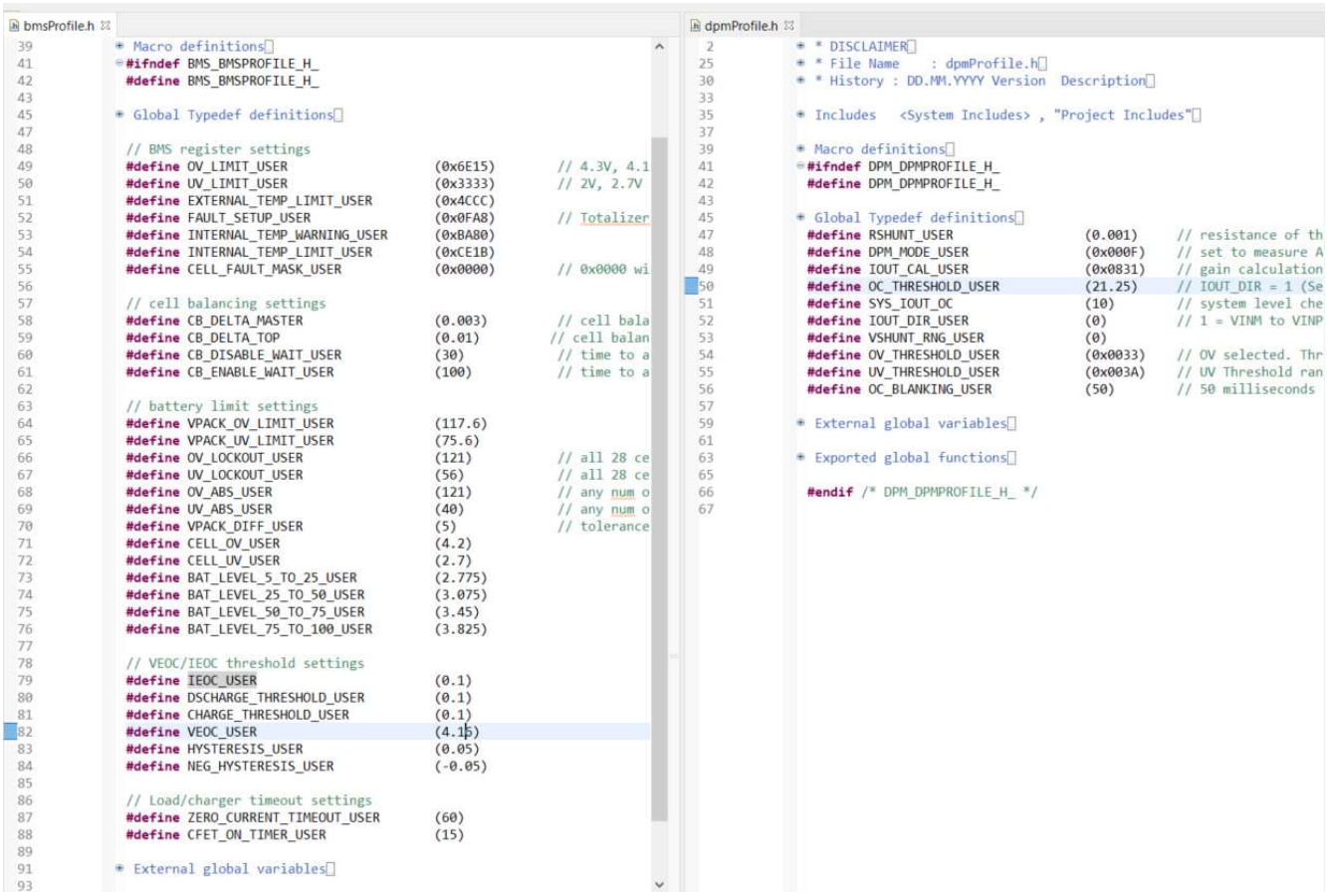


Figure 24. Profile Header Files

The user settings and their usage are outlined in Table 5 and Table 6 below. These values can be adjusted by the user to fit their end application. Note that some register settings adhere to multiple settings, such that they are not fully listed here. Refer to the datasheet for more information.

Name	Usage	Default Value
OV_LIMIT_USER	Sets the OV limit	4.3
UV_LIMIT_USER	Sets the UV limit	2.0
EXTERNAL_TEMP_LIMIT_USER	Sets the external temperature limit	70
FAULT_SETUP_USER	Sets the fault settings and other misc. settings (fault totalizer, interval, etc.)	0x0FA8
INTERNAL_TEMP_WARNING_USER	Sets the internal temperature warning level	100
INTERNAL_TEMP_LIMIT_USER	Sets the internal temperature limit	139.2
CELL_FAULT_MASK_USER	Sets which cells will have their faults masked (bit value of 0 is unmasked)	0x0000
CB_DELTA_MASTER	Cell balancing voltage delta to compare with the minimum cell (master)	0.03
CB_DELTA_TOP	Cell balancing voltage delta to compare with the minimum cell (top)	0.03
CB_DISABLE_WAIT_USER	Wait time after disabling balancing before taking readings	30
CB_ENABLE_WAIT_USER	Wait time after enabling balancing to allow for time to balance	100
NUM_CELLS_USER	Number of cells used in the system (for calculations of pack range below)	28
VPACK_OV_LIMIT_USER	Pack voltage OV limit as measured in the system	117.6
VPACK_UV_LIMIT_USER	Pack voltage UV limit as measured in the system	75.6
OV_VPACK_LOCKOUT_USER	Absolute limit for pack OV as defined by number of cells used in the system	121
UV_VPACK_LOCKOUT_USER	Absolute limit for pack UV as defined by number of cells used in the system	56
OV_ABS_USER	Absolute limit for OV that the board can support for any cell count	121
UV_ABS_USER	Absolute limit for UV that the board can support for any cell count	40
VPACK_DIFF_USER	Differential threshold for BMS and system pack voltages	5
CELL_OV_USER	OV limit for a single cell	4.2
CELL_UV_USER	UV limit for a single cell	2.7
BAT_LEVEL_5_TO_25_USER	Fuel gauge 5-25% threshold	2.775
BAT_LEVEL_25_TO_50_USER	Fuel gauge 25-50% threshold	3.075
BAT_LEVEL_50_TO_75_USER	Fuel gauge 50-75% threshold	3.45
BAT_LEVEL_75_TO_100_USER	Fuel gauge 75-100% threshold	3.825
IEOC_USER	Current end-of-charge level which sets IEOC when current falls below it	0.1
DSCHARGE_THRESHOLD_USER	Current level which indicates current is present when discharging	0.1
CHARGE_THRESHOLD_USER	Current level which indicates current is present when charging	0.1
VEOC_USER	Cell voltage threshold for setting voltage end-of-charge	4.16
HYSTERESIS_USER	Hysteresis buffer for indicating current is present when discharging	0.05
NEG_HYSTERESIS_USER	Hysteresis buffer for indicating current is present when charging	-0.05
ZERO_CURRENT_TIMEOUT_USER	Timeout in seconds for when current level is near-zero	60
CFET_ON_TIMER_USER	Timeout in seconds for turning CFET back on following discharge current	15

Table 5. BMS Profile User Settings

Name	Usage	Default Value
RSHUNT_USER	Primary shunt resistance value in Ohms	0.001
DPM_MODE_USER	Selects options for which elements to scan (voltage, current, temperature)	0x000F(All)
IOUT_CAL_USER	Current gain calibration value	0x0831
OC_THRESHOLD_USER	The OC threshold value in amperes	21.25
SYS_IOUT_OC	OC level as measured by the system (for charging current)	10
IOUT_DIR_USER	Direction of current flow (1 is negative to positive)	0
VSHUNT_RNG_USER	Primary shunt full-scale range (0 is 80mV)	0
OV_THRESHOLD_USER	OV threshold for the 12V supply to the system	12.5
UV_THRESHOLD_USER	UV threshold for the 12V supply to the system	11.4
OC_BLANKING_USER	Blanking time to catch inrush current OC in milliseconds	50

Table 6. DPM Profile user settings

System Faults Overview

This section will outline the faults which are monitored and controlled in the demo project. All these faults (except for open wire faults when using a battery emulator) will cause the system to shut down when they are either confirmed by the fault handler function, or within the function that they are detected.

When a fault occurs, a traceback function will print out the function where the fault occurred and the fault which caused the shutdown, as well as some system readings. The faults can be system-level faults or the contents of fault registers within the DPM and BMS devices. The traceback can be used with the DPM and BMS datasheets to easily find the cause of faults in the system and aid with debugging. Refer to the datasheets for these devices in the “Reference Documents” section for more information.

The function and fault flag portions of the traceback code can be seen in Figure 25 below.

```
printf("/************   SYSTEM ERROR TRACEBACK   ************/\n");
auxi_write((uint8_t *)"/************   SYSTEM ERROR TRACEBACK   ************/\r\n");
printf("\n");
auxi_write((uint8_t *)"\r\n");

switch(func)
{
    case 1:
        printf("System shutdown origin: dpm_setup()\n");
        auxi_write((uint8_t *)"System shutdown origin: dpm_setup()\r\n"); break;
    case 2:
        printf("System shutdown origin: bms_setup()\n");
        auxi_write((uint8_t *)"System shutdown origin: bms_setup()\r\n"); break;
    case 3:
        printf("System shutdown origin: dpm_update()\n");
        auxi_write((uint8_t *)"System shutdown origin: dpm_update()\r\n"); break;
    case 4:
        printf("System shutdown origin: bms_update()\n");
        auxi_write((uint8_t *)"System shutdown origin: bms_update()\r\n"); break;
    case 5:
        printf("System shutdown origin: sys_main()\n");
        auxi_write((uint8_t *)"System shutdown origin: sys_main()\r\n"); break;
    case 6:
        printf("System shutdown origin: bms_get_reg()\n");
        auxi_write((uint8_t *)"System shutdown origin: bms_get_reg()\r\n"); break;
    case 7:
        printf("System shutdown origin: sys_oc_handle()\n");
        auxi_write((uint8_t *)"System shutdown origin: sys_oc_handle()\r\n"); break;
    case 8:
        printf("System shutdown origin: sys_get_vpack()\n");
        auxi_write((uint8_t *)"System shutdown origin: sys_get_vpack()\r\n"); break;
    case 9:
        printf("System shutdown origin: sys_power_on_reset()\n");
        auxi_write((uint8_t *)"System shutdown origin: sys_power_on_reset()\r\n"); break;
    case 10:
        printf("System shutdown origin: sys_user_command()\n");
        auxi_write((uint8_t *)"System shutdown origin: sys_user_command()\r\n"); break;
    default:
        printf("Other function caused shutdown\n");
        auxi_write((uint8_t *)"Other function caused shutdown\r\n"); break;
```

```
// enter all possible faults
printf("\n");
printf("Error flag trigger:\n");
switch(err_flag)
{
    case 1:
        printf("Overcurrent detected by DPM\n");
        auxi_write((uint8_t *)"Overcurrent detected by DPM\r\n"); break;
    case 2:
        printf("BMS System Fault, CRCF | SPIF | DCHF == 1\n");
        auxi_write((uint8_t *)"BMS System Fault, CRCF | SPIF | DCHF == 1\r\n"); break;
    case 3:
        printf("DPM System Fault, I2CF | PMF | VCCF | OCF == 1\n");
        auxi_write((uint8_t *)"DPM System Fault, I2CF | PMF | VCCF | OCF == 1\r\n"); break;
    case 4:
        printf("BMS UVF | OVF | OWF | OTF == 1\n");
        auxi_write((uint8_t *)"BMS UVF | OVF | OWF | OTF == 1\r\n"); break;
    case 5:
        printf("DPM IOUTF | VOUTF | TEMPF == 1\n");
        auxi_write((uint8_t *)"DPM IOUTF | VOUTF | TEMPF == 1\r\n"); break;
    case 6:
        printf("Vpack is OV or UV\n");
        auxi_write((uint8_t *)"Vpack is OV or UV\r\n"); break;
    case 7:
        printf("System vpack and BMS vpack differential is too large\n");
        auxi_write((uint8_t *)"System vpack and BMS vpack differential is too large\r\n"); break;
    case 8:
        printf("Vpack over/under OV/UV lockout threshold - bad battery\n");
        auxi_write((uint8_t *)"pack over/under OV/UV lockout threshold - bad battery\r\n"); break;
    case 9:
        printf("No error- user called shutdown\n");
        auxi_write((uint8_t *)"No error - user called shutdown\r\n"); break;
    default:
        printf("No error encountered\n");
        auxi_write((uint8_t *)"No error encountered\r\n"); break;
}
```

Figure 25. System Traceback Code

Table 7 below shows the system faults for each module which are defined by the demo project separately from device register faults.

Fault Bit	Cause	Fault Variable
BMSF	Fault asserted by BMS register such as OW, OV, UV, etc.	g_bms_fault
CRCF	CRC validation failed during read operation	g_bms_fault
SPIF	SPI communication fault during initial test in setup routine	g_bms_fault
DHCF	Daisy chain was not established for 2 devices correctly	g_bms_fault
DPMF	Fault asserted by DPMregister such as OC, OV, UV, etc.	g_dpm_fault
I2CF	I2C communication fault during initial test in setup routine	g_dpm_fault
PMF	12V supply is either OV or UV	g_dpm_fault
VCCF	3.3V supply is out of range	g_dpm_fault
OCF	System check for overcurrent	g_dpm_fault
AUXF	Fault asserted in the Auxi module	g_auxi_fault
UARTF	UART communication fault occurred during UART write	g_auxi_fault
BMSF	BMS system fault (duplicate)	g_sys_fault
DPMF	DPM system fault (duplicate)	g_sys_fault
AUXF	Auxi system fault (duplicate)	g_sys_fault
VBATF	Voltage pack range fault (differential, abs. limits, etc.)	g_sys_fault

Table 7. System-level Faults

Faults which are based on the BMS and DPM device fault registers are shown below in Table 8. Refer to the datasheets for the devices for further details.

Fault Register	Cause	Fault Variable
Fault Status	Conditions such as UV, OV, OW, OT, etc.	g_bms_xx.faultStatus
OV Status	Overvoltage for cells 1-14	g_bms_xx.OVFault
UV Status	Undervoltage for cells 1-14	g_bms_xx.UVFault
OW Status	Open wire fault for cells 1-14	g_bms_xx.OWFault
OT Status	Overtemperature for internal temperature of external temps 1-4	g_bms_xx.OTFault
Status Word	Hierarchical register for voltage, current, and temperature faults	g_dpm_dpmFault
Vout Status	Undervoltage and overvoltage faults	g_dpm_voutStatus

Table 8. Device Register Faults

Schematics

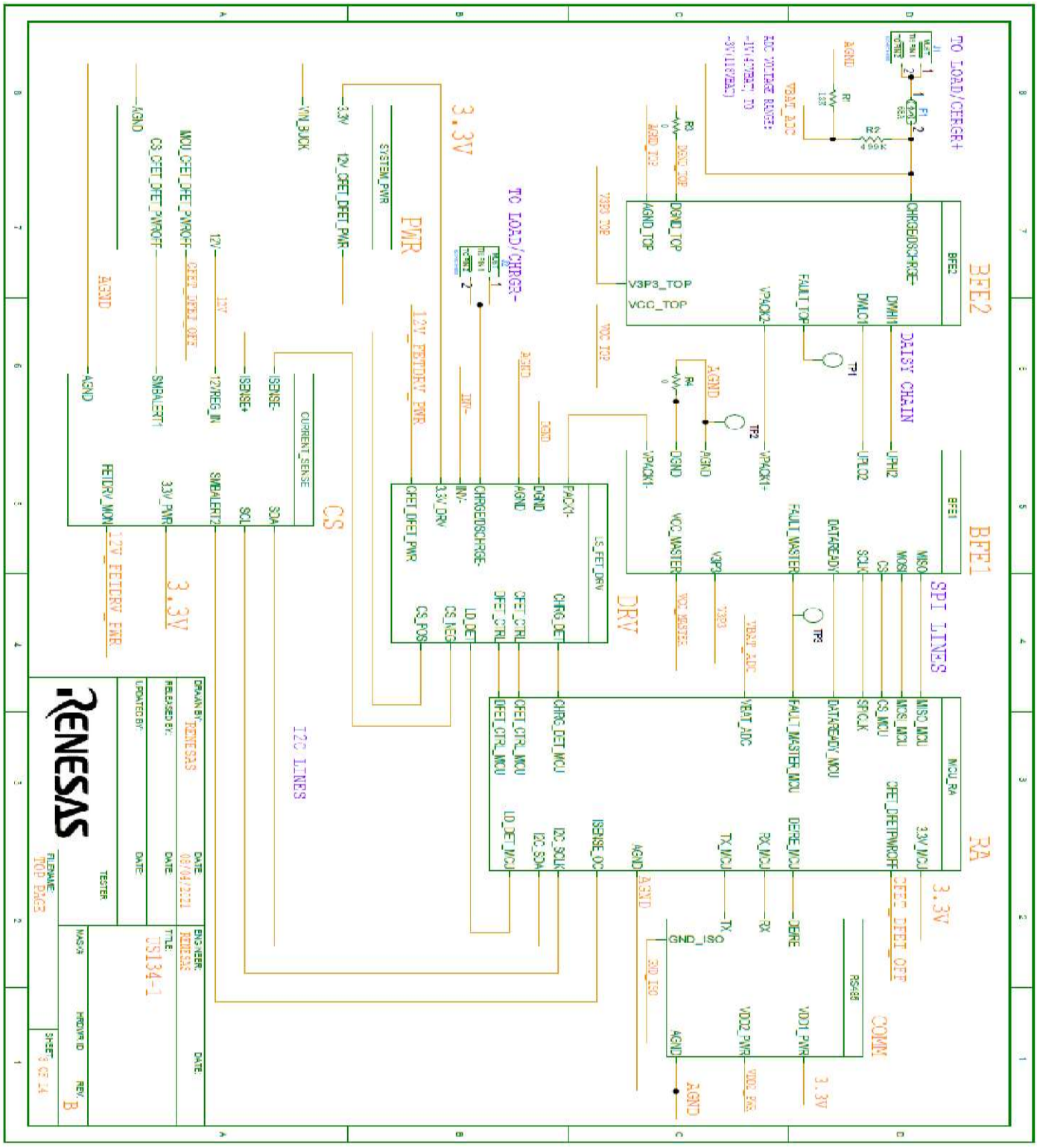


Figure 26. Schematic Top Page

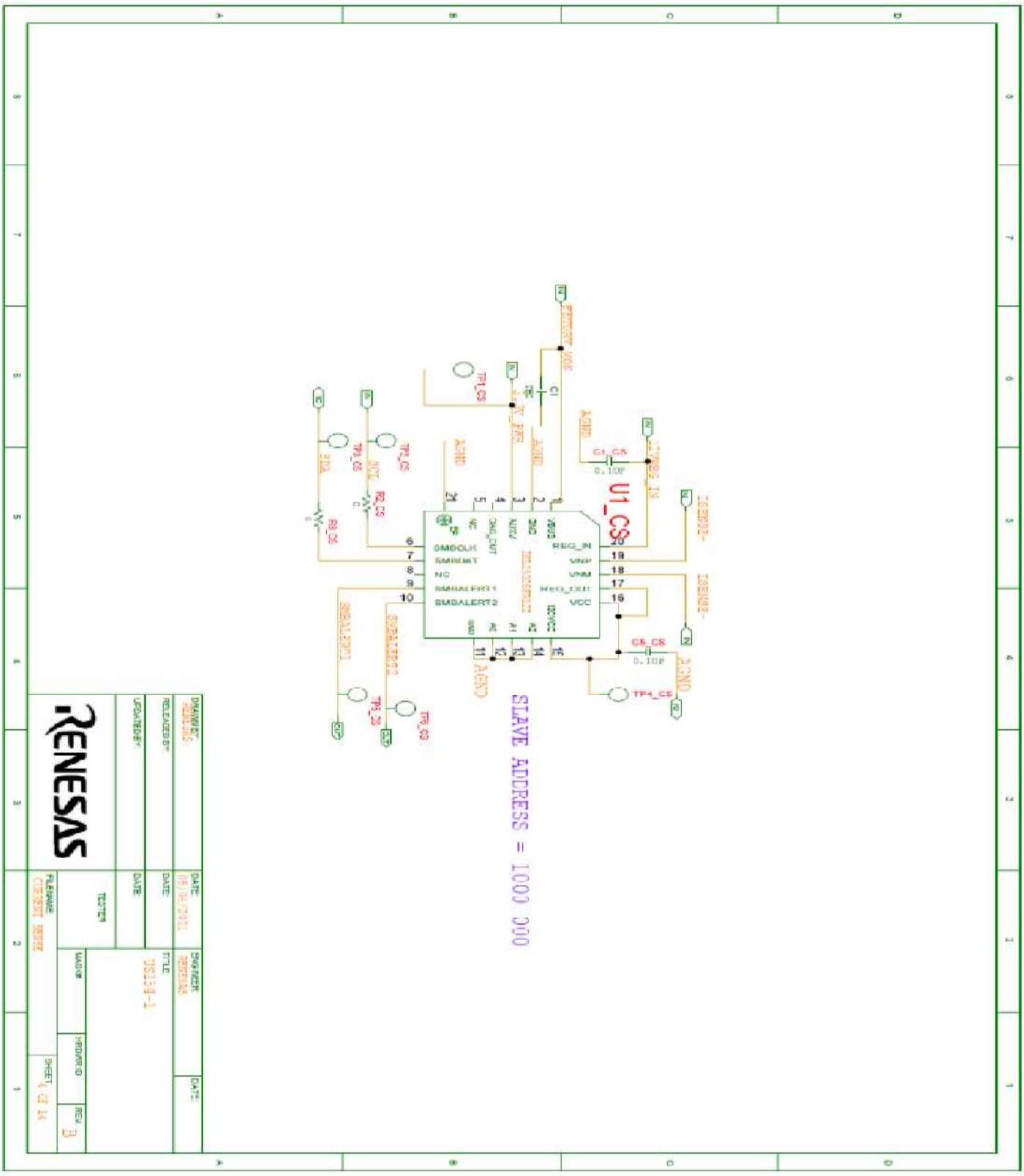


Figure 27. Schematic Current Sense

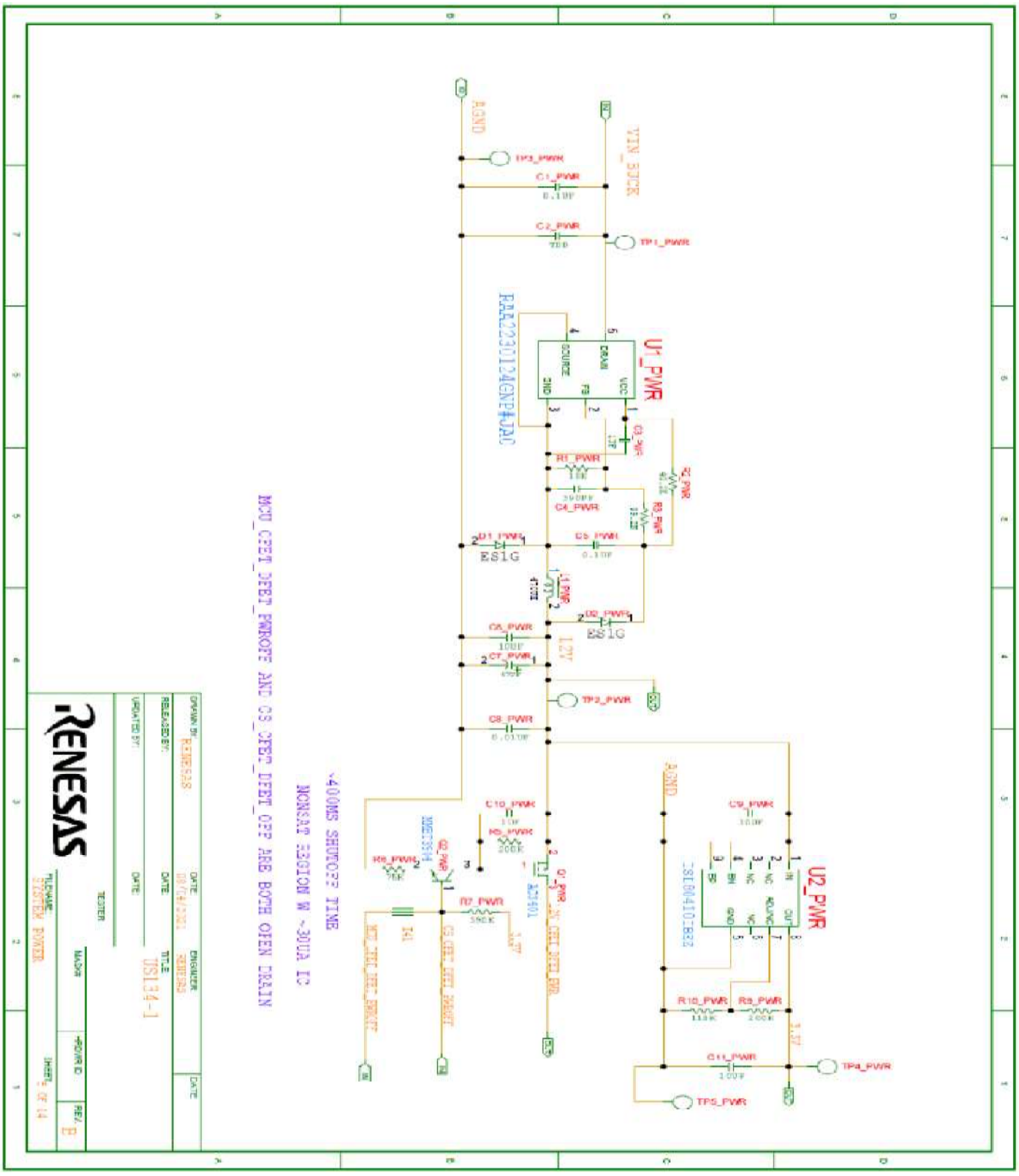


Figure 28. Schematic System Power

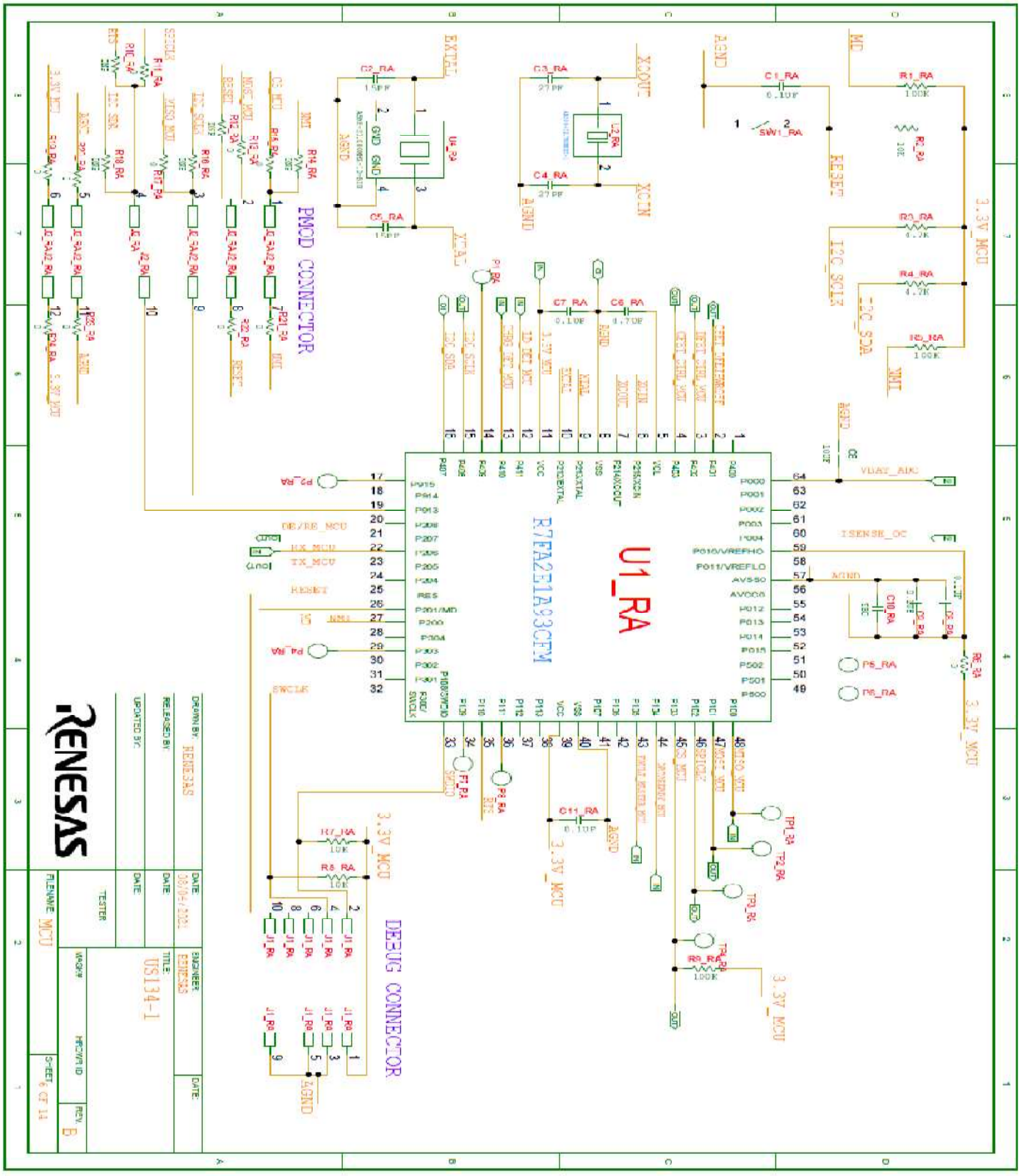


Figure 29. Schematic MCU

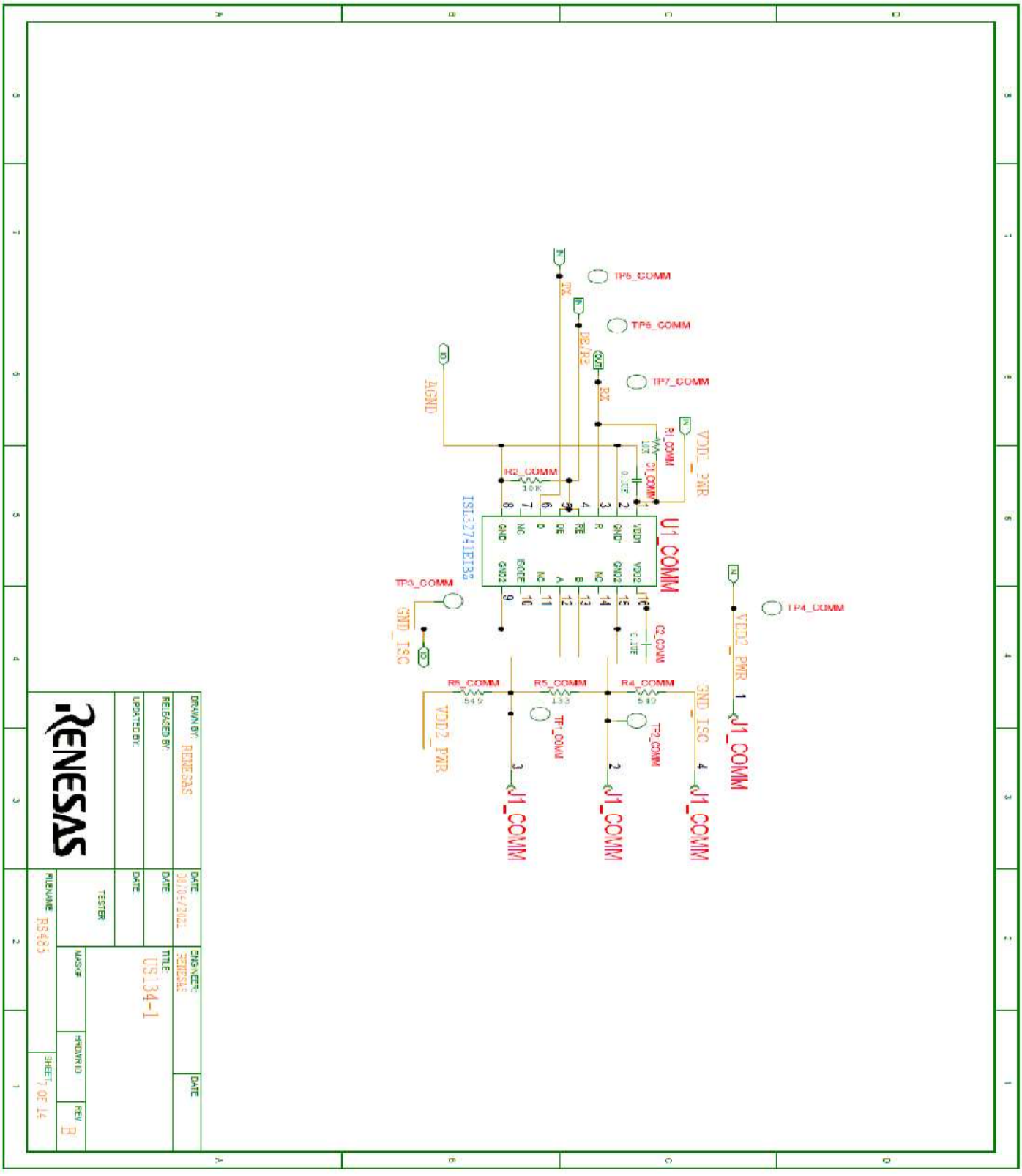


Figure 30. Schematic RS485

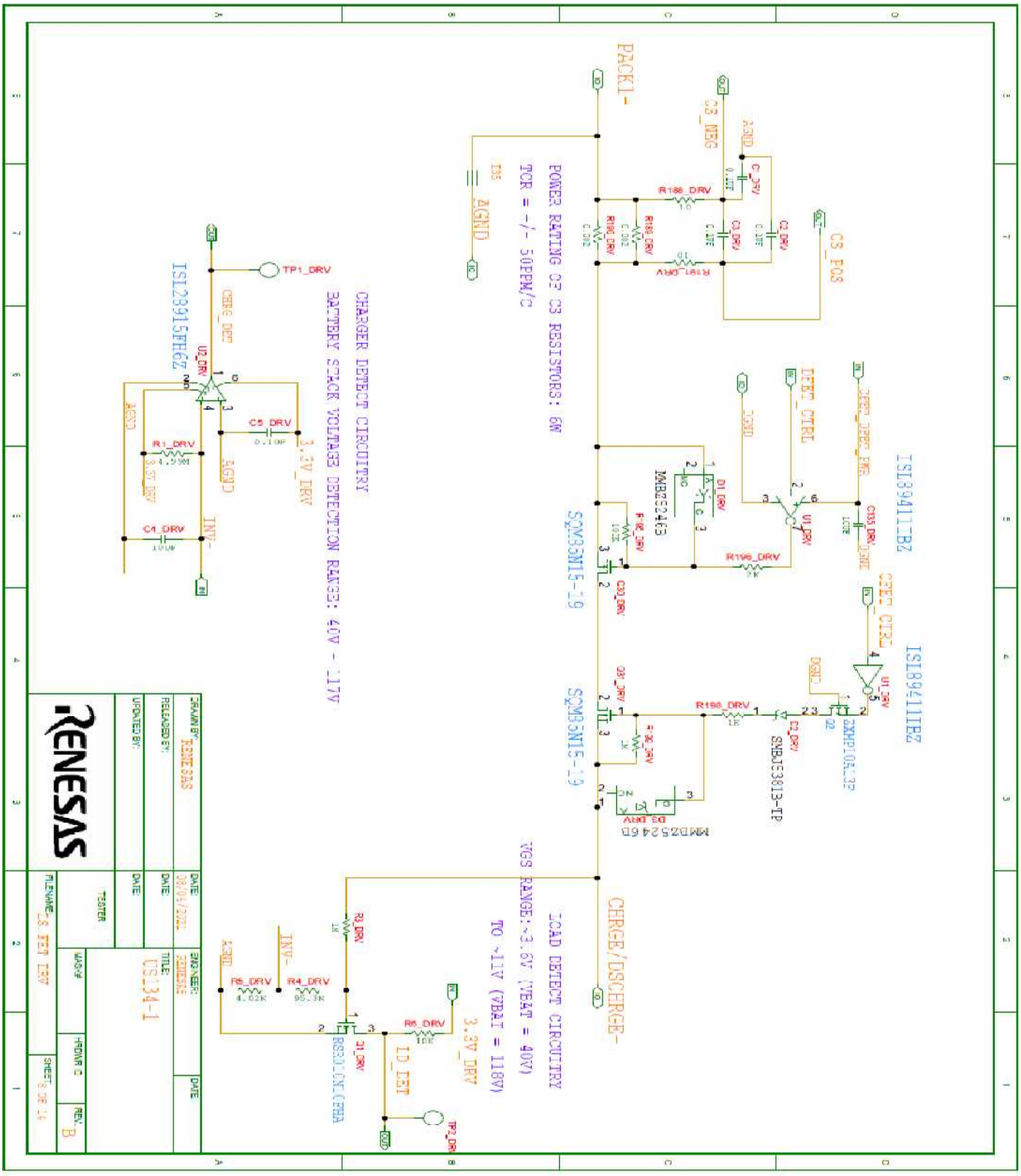


Figure 31. Schematic LS FET DRV

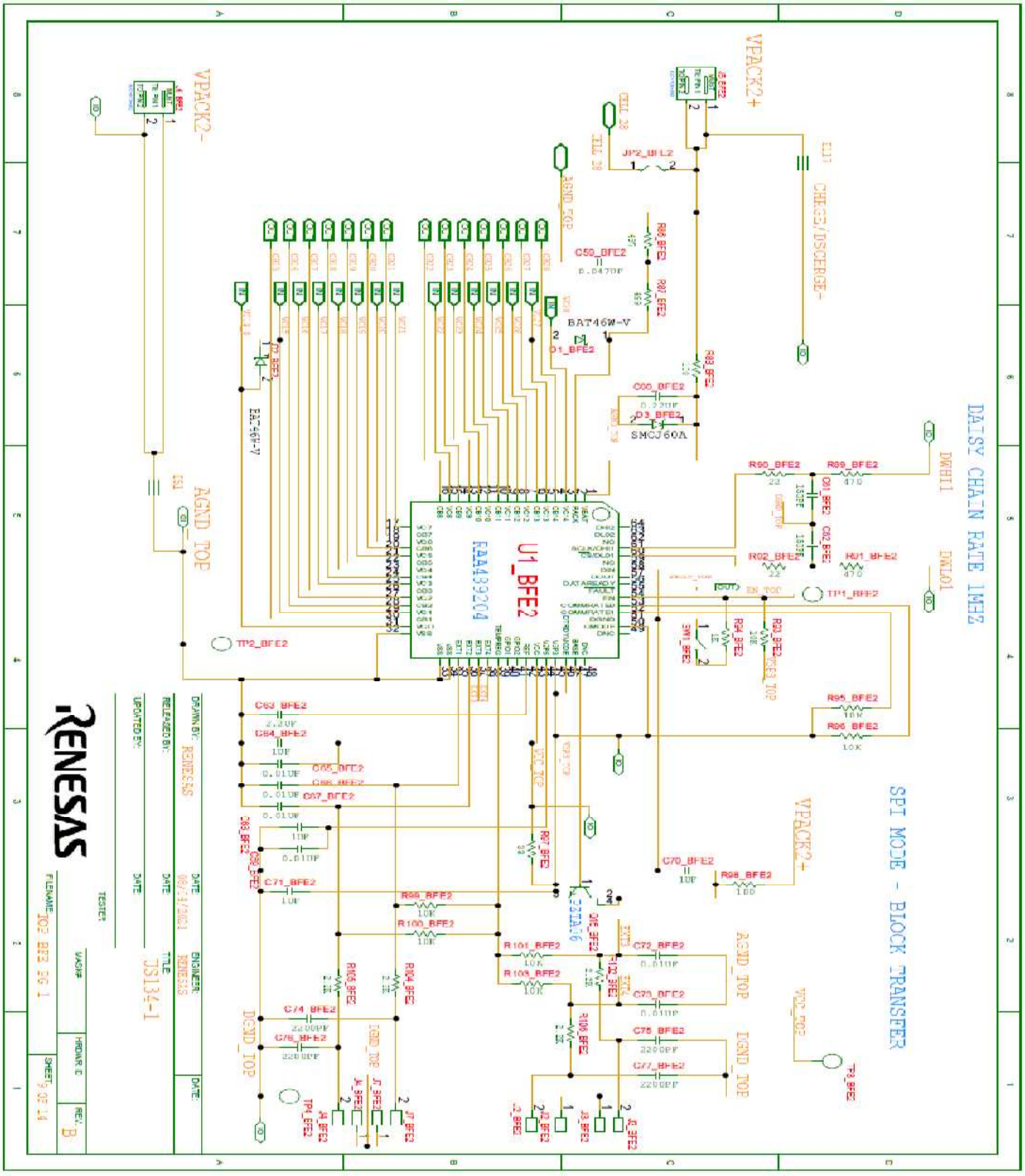


Figure 32. Schematic Top BFE Page 1

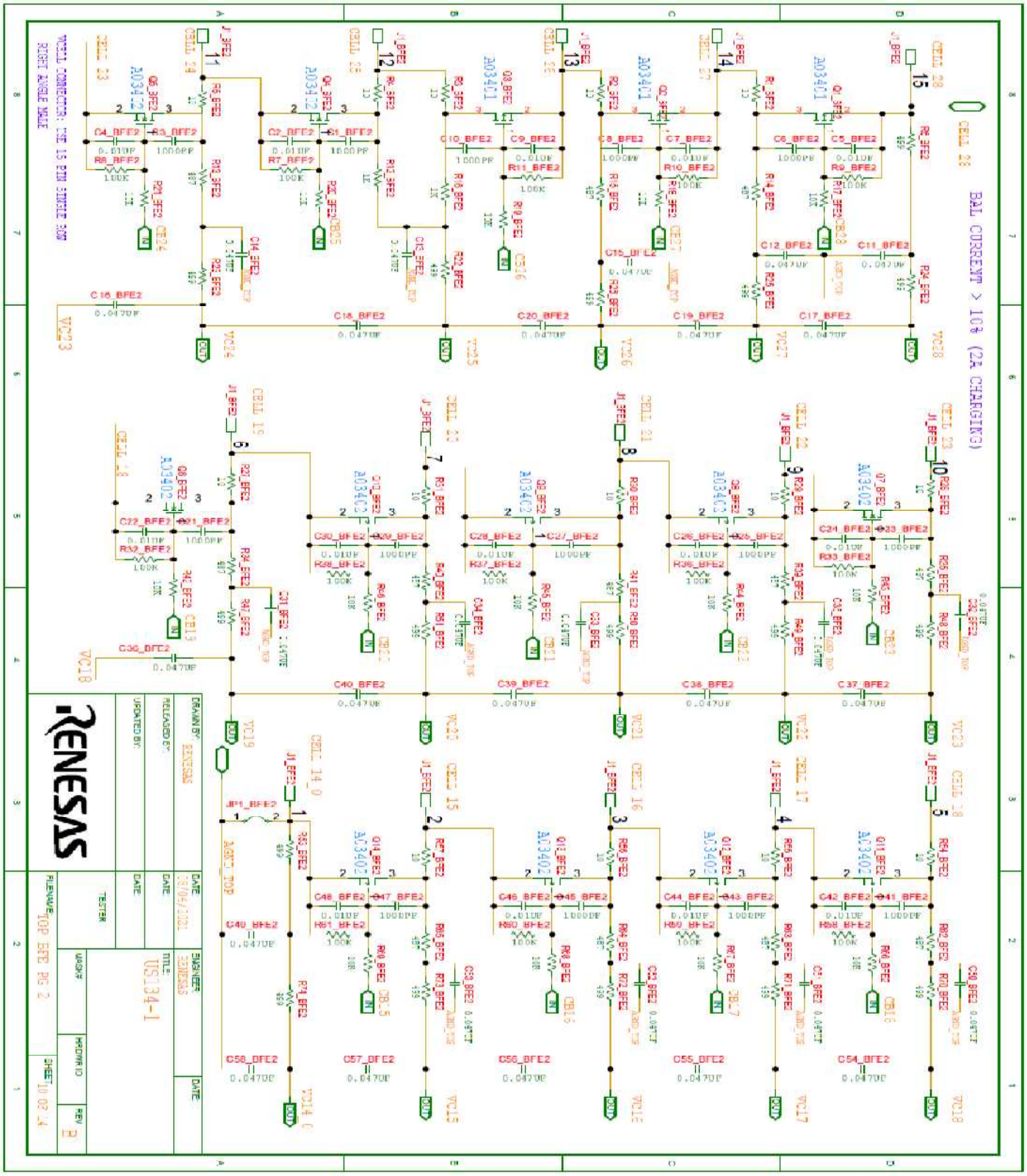


Figure 33. Schematic Top BFE Page 2

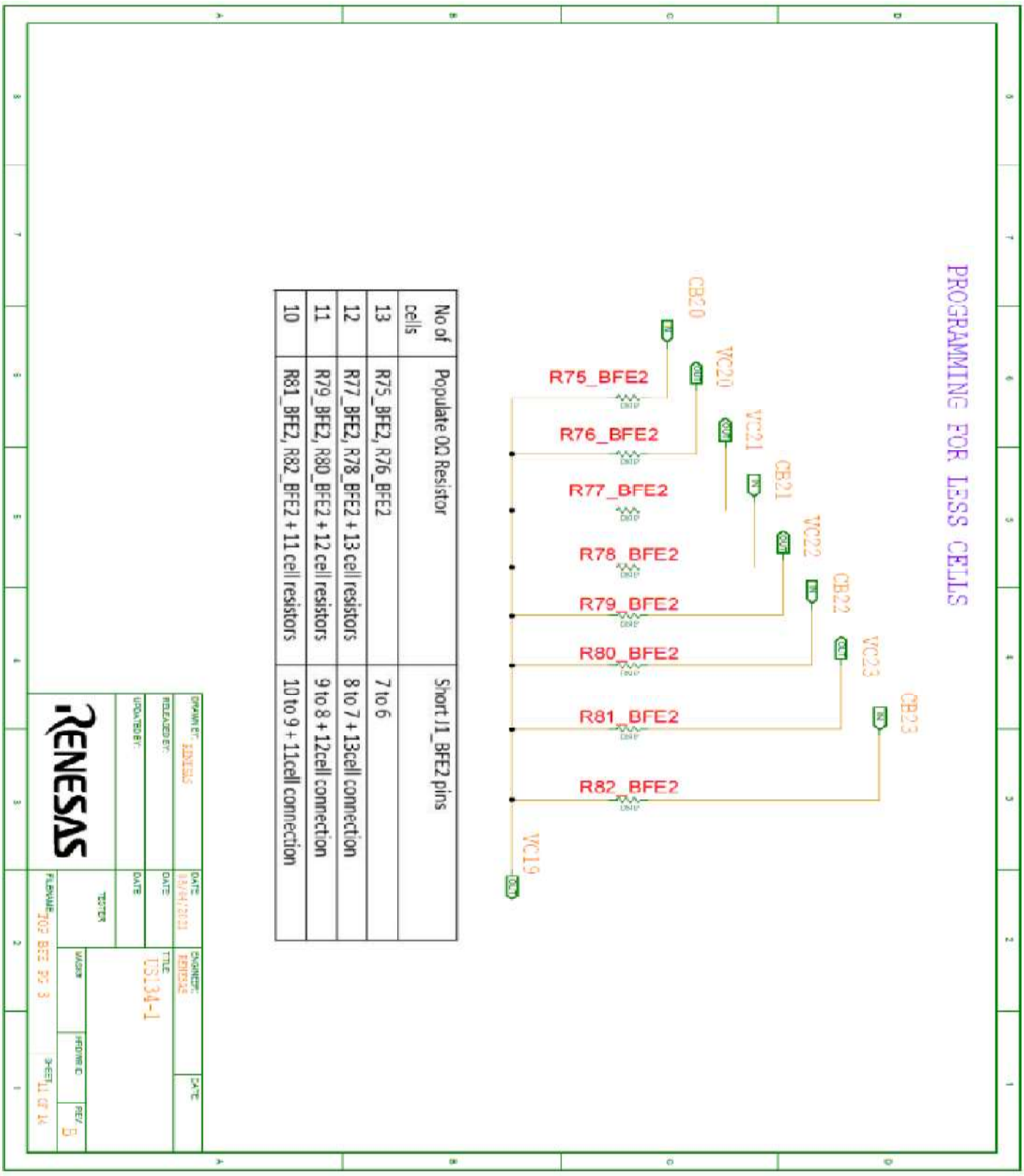


Figure 34. Schematic Top BFE Page 3

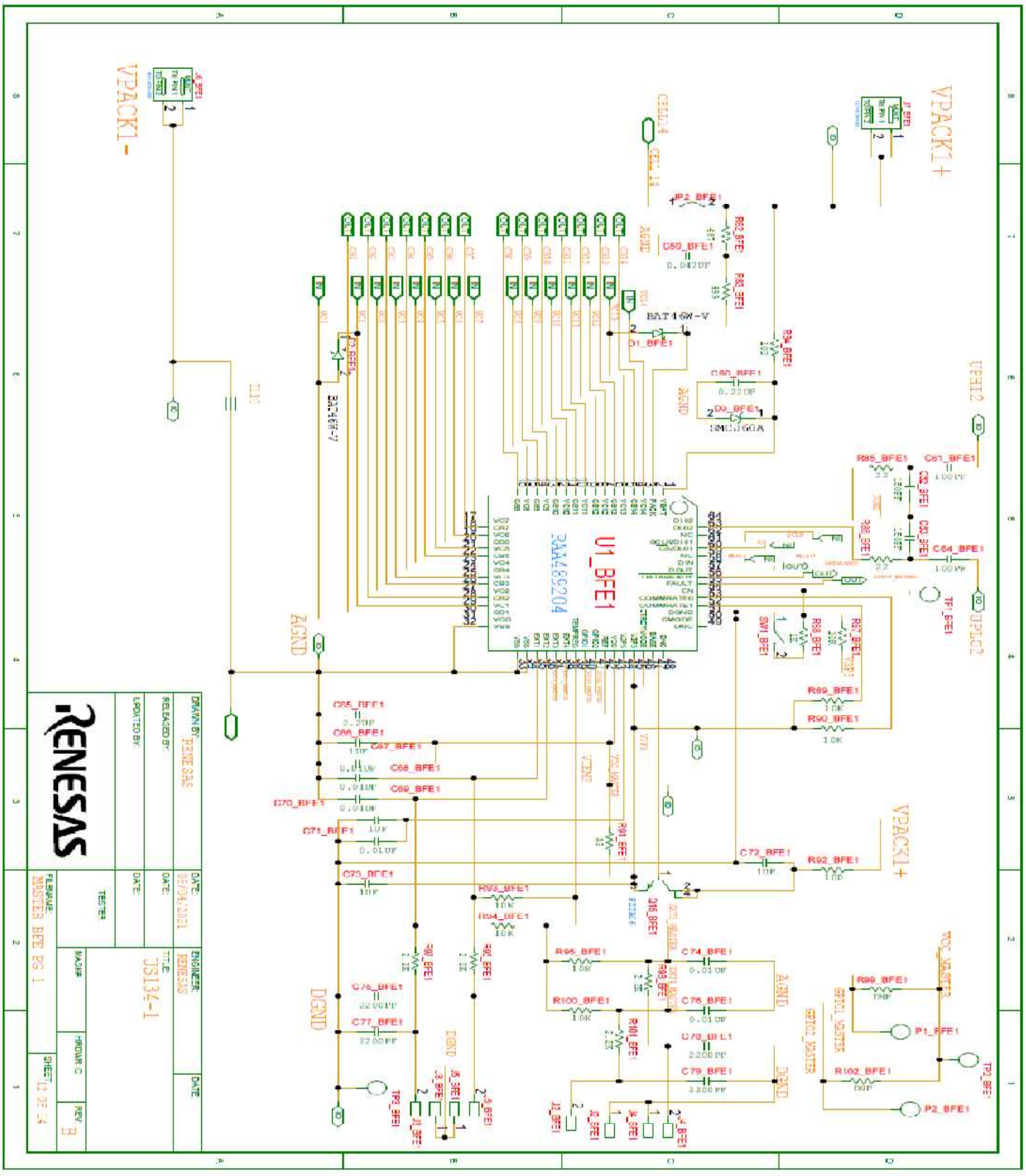


Figure 35. Schematic Master BFE Page 1

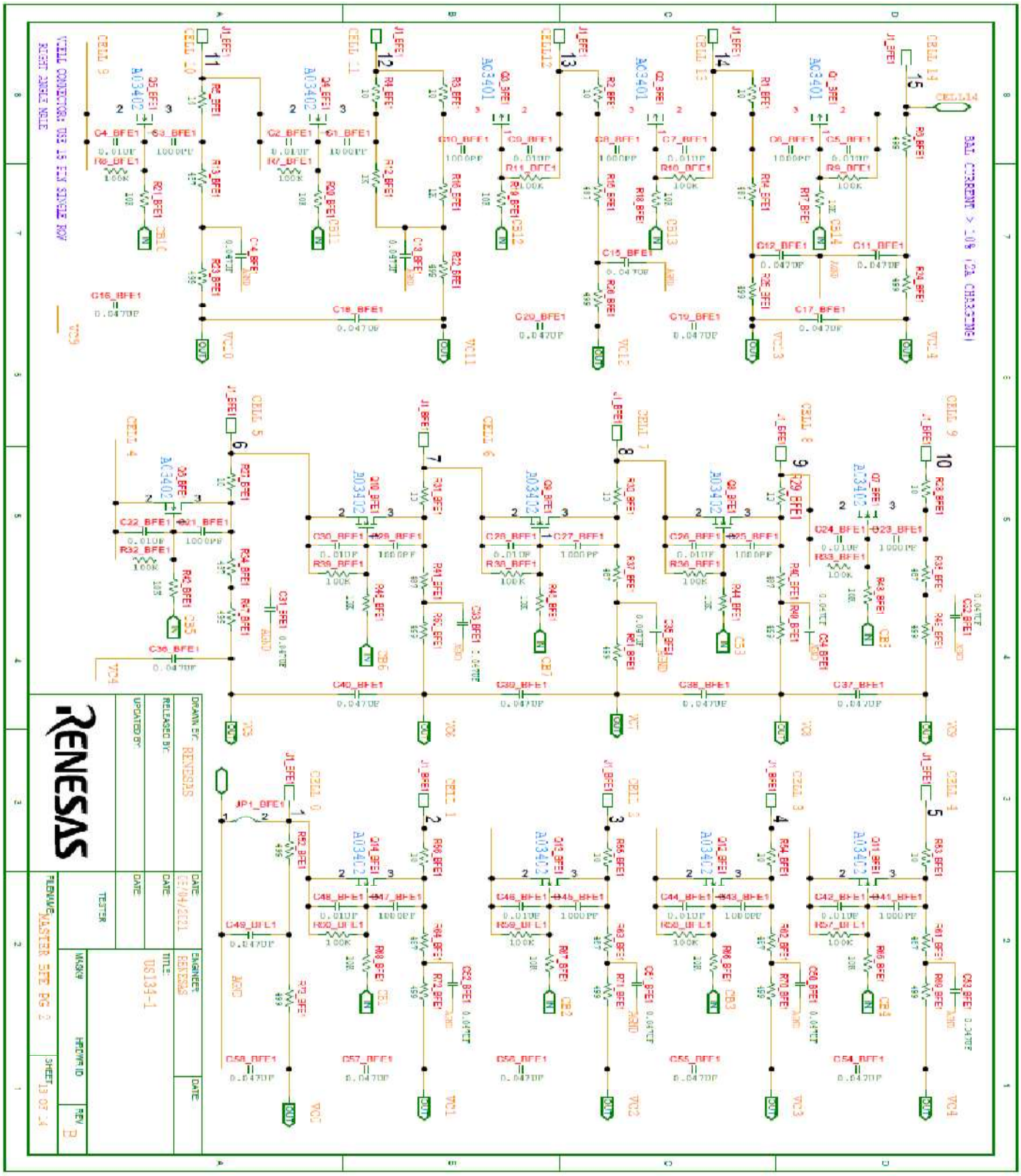


Figure 36. Schematic Master BFE Page 2

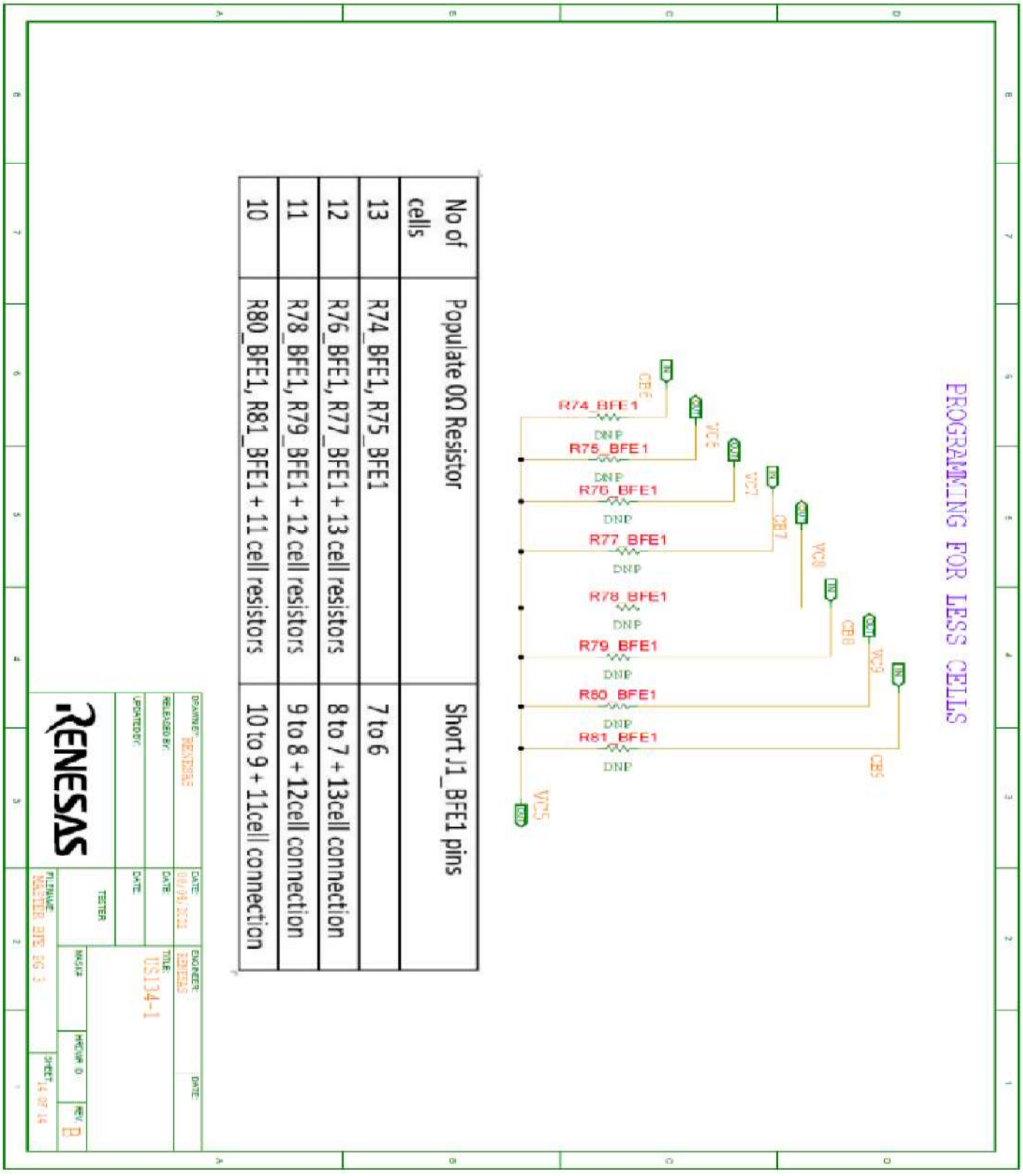


Figure 37. Schematic Master BFE Page 3

BOM

Part Number	Ref Des	Qty	Value	Package Type	Description
1725672	J1_COMM	1		4POS	4 Position @ 100 Mil Micro-Pitch Terminal Block
352210RJT	R1_BFE1,R1_BFE2,R2_BFE1,R2_BFE2,R3_BFE1,R3_BFE2,R4_BFE1,R4_BFE2,R5_BFE1,R5_BFE2,R27_BFE1,R27_BFE2,R28_BFE1,R28_BFE2,R29_BFE1,R29_BFE2,R30_BFE1,R30_BFE2,R31_BFE1,R31_BFE2,R53_BFE1,R54_BFE1,R54_BFE2,R55_BFE1,R55_BFE2,R56_BFE1,R56_BFE2,R57_BFE2	28	10	2512	Surface Mount Power Resistor
5001	TP1-TP3,TP1_CS-TP6_CS,TP1_RA-TP4_RA,TP1_DRV,TP1_PWR-TP5_PWR,TP2_DRV,TP1_BFE1,TP1_BFE2,TP1_COMM-TP7_COMM,TP2_BFE1,TP2_BFE2,TP3_BFE1,TP3_BFE2,TP4_BFE2	34		THOLE	Miniature Black Test Point .100 Pad .040 Thole
PEC36SBAN	J1_BFE1,J1_BFE2	2		15	CONN HEADER R/A 36POS 2.54MM
A03401	Q1_PWR,Q1_BFE1,Q1_BFE2,Q2_BFE1,Q2_BFE2,Q3_BFE1,Q3_BFE2	7		SOT23	P-Channel 30V 4A Rds=50mohm MOSFET
A03402	Q4_BFE1,Q4_BFE2,Q5_BFE1,Q5_BFE2,Q6_BFE1,Q6_BFE2,Q7_BFE1,Q7_BFE2,Q8_BFE1,Q8_BFE2,Q9_BFE1,Q9_BFE2,Q10_BFE1,Q10_BFE2,Q11_BFE1,Q11_BFE2,Q12_BFE1,Q12_BFE2,Q13_BFE1,Q13_BFE2,Q14_BFE1,Q14_BFE2	22		SOT23	N-Channel 30V 4A Rds=52mohm MOSFET
ABM8-20.000MHZ-10-B1U	U4_RA	1		SMD4	20.000Mz 10ppm 4-SMD Crystal
ABS06-32.768KHZ-1	U2_RA	1		SMD	32.768Khz SMD Low Profile Crystal
AUTO-CAP-SM1206-TBD	C2_PWR	1	TBD	1206	Ceramic Chip Cap (Automotive AEC-Q200)
B2B-XH-A	J2_BFE1,J2_BFE2,J3_BFE1,J3_BFE2,J4_BFE1,J4_BFE2,J5_BFE1,J7_BFE2	8		2	Header 1 Row 2 Pin @ .1 inch Pitch Single Row Vertical PTH Connector
DNP	J5_BFE2, J6_BFE2, J7_BFE1, J6_BFE1	6	DNP	DNP	DNP

B2C-PCB-HEX	J1,J2	6		B2C	Single - Hex Screw - .140in pcb depth Screw Down Large Wire Type Power Terminal
B3U-1000P	SW1_RA,SW1_BFE1,SW1_BFE2	3		SPST	SPST SMT Ultra-small Push-Button Switch
BAT46W-V	D1_BFE1,D1_BFE2,D2_BFE1,D2_BFE2	4		SOD123	100V 150ma 150mw SMALL SIGNAL SCHOTTKY DIODE
C0603C101F1GACTU	C61_BFE1,C64_BFE1	2	100PF	0603	Multilayer Capacitor
C0603C473K1RACTU	C59_BFE1,C59_BFE2	2	0.047UF	0603	MULTILAYER CAP
C0805C224K1RACAUTO	C60_BFE1,C60_BFE2	2	0.22UF	0805	Ceramic Chip Capacitor (Automotive AEC-Q200)
C1608X7R1H104K	C5_PWR	1	0.1UF	0603	MULTILAYER CAP
C1608X7S2A473K	C11_BFE1,C11_BFE2,C12_BFE1,C12_BFE2,C13_BFE1,C13_BFE2,C14_BFE1,C14_BFE2,C15_BFE1,C15_BFE2,C16_BFE1,C16_BFE2,C17_BFE1,C17_BFE2,C18_BFE1,C18_BFE2,C19_BFE1,C19_BFE2,C20_BFE1,C20_BFE2,C31_BFE1,C31_BFE2,C32_BFE1,C32_BFE2,C33_BFE1,C33_BFE2,C34_BFE1,C34_BFE2,C35_BFE1,C35_BFE2,C36_BFE1,C36_BFE2,C37_BFE1,C37_BFE2,C38_BFE1,C38_BFE2,C39_BFE1,C39_BFE2,C40_BFE1,C40_BFE2,C49_BFE1,C49_BFE2,C50_BFE1,C50_BFE2,C51_BFE1,C51_BFE2,C52_BFE1,C52_BFE2,C53_BFE1,C53_BFE2,C54_BFE1,C54_BFE2,C55_BFE1,C55_BFE2,C56_BFE1,C56_BFE2,C57_BFE1,C57_BFE2,C58_BFE1,C58_BFE2	60	0.047UF	0603	MULTILAYER CAP
C3216X7R2E104K160AA	C1_PWR	1	0.1UF	1206	Multilayer Cap
CC0402JRNPO9BN150	C2_RA,C5_RA	2	15PF	0402	Ceramic Capacitor
CC0402JRNPO9BN270	C3_RA,C4_RA	2	27PF	0402	Ceramic Capacitor
CC0603FRNPO0BN151	C61_BFE2,C62_BFE1,C62_BFE2,C63_BFE1	4	150PF	0603	Multilayer Capacitor
CRCW06030000Z0EA	R11_RA,R13_RA,R15_RA,R17_RA,R19_RA-R24_RA	10	0	0603	Film Chip Resistor (Automotive AEC-Q200)
CRCW060322R0FKEA	R85_BFE1,R86_BFE1,R90_BFE2,R92_BFE2	4	22	0603	Thick Film Chip Resistor (Automotive AEC-Q200)

CRCW06034 70RFKEA	R89_BFE2,R91_BFE2	2	470	0603	Thick Film Chip Resistor
CRCW06034 M99FKEA	R1_DRV	1	4.99M	0603	Thick Film Chip Resistor
CRGCQ0603 F33R	R91_BFE1,R97_BFE2	2	33	0603	Moisture Resistant Thick Film Resistor
CSRL3-0R002F8	R189_DRV,R190_DRV	2	0.002	2512	Current Sense Metal Element Resistor
EMVA250A DA470MF55 G	C7_PWR	1	47UF	SMD	60mA ALUMINUM ELECTROLYTIC CAPACITOR
ERA-3AEB5490V	R4_COMM,R6_COMM	2	549	0603	Film Chip Resistor
ERJ-14NF1000U	R92_BFE1,R98_BFE2	2	100	1210	Thick Film Chip Resistor (Automotive AEC-Q200)
ERJ-3EKF2001V	R196_DRV	1	2K	0603	Thick Film Chip Resistor
ERJ-P6WF1000V	R84_BFE1,R88_BFE2	2	100	0805	Thick Film Chip Resistor (Automotive AEC-Q200)
ES1G	D1_PWR,D2_PWR	2		DO214	1A 400V Fast Rectifier Diode
FTSH-105-01-L-DV-007	J1_RA	1		SINGLE-7	10 Pin 2 X 5 @ 50 mil pitch SMT Header Connector
GCJ188R71 H104KA12D	C1_RA,C7_RA-C9_RA,C11_RA,C1_COMM,C2_COM M	7	0.1UF	0603	Multilayer Cap (Automotive AEC-Q200)
GCM188R7 1E105KA64 D	C64_BFE2,C66_BFE1,C68_BFE2,C70_ BFE1,C71_BFE2,C73_BFE1	6	1UF	0603	Ceramic Chip Cap
GRM155R7 1E103KA01	C2_BFE1,C2_BFE2,C4_BFE1,C4_BFE2, C5_BFE1,C5_BFE2,C7_BFE1,C7_BFE2, C9_BFE1,C9_BFE2,C22_BFE1,C22_BF E2,C24_BFE1,C24_BFE2,C26_BFE1,C2 6_BFE2,C28_BFE1,C28_BFE2,C30_BFE 1,C30_BFE2,C42_BFE1,C42_BFE2,C44 _BFE1,C44_BFE2,C46_BFE1,C46_BFE2 ,C48_BFE1,C48_BFE2	28	0.01UF	0402	Multilayer Cap
GRM188R7 2A104KA35J	C1_DRV-C3_DRV	3	0.1UF	0603	Multilayer Cap
GRM32ER7 1H106KA12 L	C6_PWR	1	10UF	1210	Ceramic Chip Cap
H1044-00102-50V10	C1_BFE1,C1_BFE2,C3_BFE1,C3_BFE2, C6_BFE1,C6_BFE2,C8_BFE1,C8_BFE2, C10_BFE1,C10_BFE2,C21_BFE1,C21_ BFE2,C23_BFE1,C23_BFE2,C25_BFE1,	28	1000PF	0402	Multilayer Cap

	C25_BFE2,C27_BFE1,C27_BFE2,C29_BFE1,C29_BFE2,C41_BFE1,C41_BFE2,C43_BFE1,C43_BFE2,C45_BFE1,C45_BFE2,C47_BFE1,C47_BFE2				
H1044-00222-25V10	C74_BFE2,C75_BFE1,C75_BFE2,C76_BFE2,C77_BFE1,C77_BFE2,C78_BFE1,C79_BFE1	8	2200PF	0402	Multilayer Cap
H1045-00103-16V10	C65_BFE2,C67_BFE1,C69_BFE2,C71_BFE1	4	0.01UF	0603	Multilayer Cap
H1045-00103-25V10	C66_BFE2,C67_BFE2,C68_BFE1,C69_BFE1,C72_BFE2,C73_BFE2,C74_BFE1,C76_BFE1	8	0.01UF	0603	Multilayer Cap
H1045-00104-50V10	C1_CS,C5_CS,C5_DRV	3	0.1UF	0603	Multilayer Cap
H1045-00391-50V5	C4_PWR	1	390PF	0603	Multilayer Cap
H1045-00475-10V10-T	C6_RA	1	4.7UF	0603	Multilayer Cap
H1045-TBD-TBD	C10_RA	1	TBD	0603	Multilayer Cap
H1046-00103-50V10	C8_PWR	1	0.01UF	0805	Multilayer Cap
H1065-00105-100V10	C70_BFE2,C72_BFE1	2	1UF	1206	Multilayer Cap
H2505-02201-1/16WR1	R96_BFE1,R97_BFE1,R98_BFE1,R101_BFE1,R102_BFE2,R104_BFE2,R105_BFE2,R106_BFE2	8	2.2K	0603	Metal Film Chip Resistor
H2505-DNP-DNP-1	R10_RA,R12_RA,R14_RA,R16_RA,R18_RA,R74_BFE1,R75_BFE1,R75_BFE2,R76_BFE1,R76_BFE2,R77_BFE1,R77_BFE2,R78_BFE1,R78_BFE2,R79_BFE1,R79_BFE2,R80_BFE1,R80_BFE2,R81_BFE1,R81_BFE2,R82_BFE2	21	DNP	0603	Metal Film Chip Resistor (Do Not Populate)
H2510-01002-1/16W1	R17_BFE1,R17_BFE2,R18_BFE1,R18_BFE2,R19_BFE1,R19_BFE2,R20_BFE1,R20_BFE2,R21_BFE1,R21_BFE2,R42_BFE1,R42_BFE2,R43_BFE1,R43_BFE2,R44_BFE1,R44_BFE2,R45_BFE1,R45_BFE2,R46_BFE1,R46_BFE2,R65_BFE1,R66_BFE1,R66_BFE2,R67_BFE1,R67_BFE2,R68_BFE1,R68_BFE2,R69_BFE2,	32	10K	0402	Thick Film Chip Resistor

	R89_BFE1,R90_BFE1,R95_BFE2,R96_BFE2				
H2510-01003-1/16W1	R7_BFE1,R7_BFE2,R8_BFE1,R8_BFE2,R9_BFE1,R9_BFE2,R10_BFE1,R10_BFE2,R11_BFE1,R11_BFE2,R32_BFE1,R32_BFE2,R33_BFE1,R33_BFE2,R36_BFE1,R36_BFE2,R37_BFE2,R38_BFE1,R38_BFE2,R39_BFE1,R57_BFE1,R58_BFE1,R58_BFE2,R59_BFE1,R59_BFE2,R60_BFE1,R60_BFE2,R61_BFE2	28	100K	0402	Thick Film Chip Resistor
H2510-DNP-DNP-1	R99_BFE1,R102_BFE1	2	DNP	0402	Thick Film Chip Resistor(Do Not Populate)
H2511-00R00-1/10W1	R2_CS,R3_CS,R6_RA	3	0	0603	Thick Film Chip Resistor
H2511-01001-1/16W1	R12_BFE1,R12_BFE2,R16_BFE1,R16_BFE2,R94_BFE2, R198_DRV	6	1K	0603	Thick Film Chip Resistor
H2511-01002-1/10W1	R2_RA,R7_RA,R8_RA,R1_PWR,R6_DRV,R1_COMM,R2_COMM,R87_BFE1,R93_BFE1,R93_BFE2,R94_BFE1,R95_BFE1,R99_BFE2,R100_BFE1,R100_BFE2,R101_BFE2,R103_BFE2	17	10K	0603	Thick Film Chip Resistor
H2511-01003-1/10W1	R1_RA,R5_RA,R9_RA,R195_DRV	4	100K	0603	Thick Film Chip Resistor
H2511-01004-1/16W1	R3_DRV, R199_DRV	2	1M	0603	Thick Film Chip Resistor
H2511-01183-1/16W1	R10_PWR	1	118K	0603	Thick Film Chip Resistor
H2511-01330-1/16W1	R5_COMM	1	133	0603	Thick Film Chip Resistor
H2511-02003-1/10W1	R5_PWR,R9_PWR	2	200K	0603	Thick Film Chip Resistor
H2511-03903-1/10W1	R7_PWR	1	390K	0603	Thick Film Chip Resistor

H2511-03922-1/16W1	R3_PWR	1	39.2K	0603	Thick Film Chip Resistor
H2511-04021-1/16W1	R5_DRV	1	4.02K	0603	Thick Film Chip Resistor
H2511-04022-1/10W1	R2_PWR	1	40.2K	0603	Thick Film Chip Resistor
H2511-04870-1/10W1	R13_BFE1,R13_BFE2,R14_BFE1,R14_BFE2,R15_BFE1,R15_BFE2,R34_BFE1,R34_BFE2,R35_BFE1,R35_BFE2,R37_BFE1,R39_BFE2,R40_BFE1,R40_BFE2,R41_BFE1,R41_BFE2,R61_BFE1,R62_BFE1,R62_BFE2,R63_BFE1,R63_BFE2,R64_BFE1,R64_BFE2,R65_BFE2,R82_BFE1,R86_BFE2	26	487	0603	Thick Film Chip Resistor
H2511-04990-1/16W1	R6_BFE1,R6_BFE2,R52_BFE1,R53_BFE2	4	499	0603	Thick Film Chip Resistor
H2511-07502-1/10W1	R6_PWR	1	75K	0603	Thick Film Chip Resistor
H2511-09532-1/16W1	R4_DRV	1	95.3K	0603	Thick Film Chip Resistor
H2513-00R00-1/8W1	R3,R4	2	0	1206	Thick Film Chip Resistor
ISL28025FR12Z	U1_CS	1		20QFN	Precision Digital Power Monitor with Real Time Alerts
ISL28915FH6Z	U2_DRV	1		SOT23-6A	Nano Power Push/Pull Output Comparator
ISL32741EIBZ	U1_COMM	1		16SOICW-2	6kV VDE-Reinforced Isolated 40Mbps RS-485 Transceiver
ISL80410IBEZ	U2_PWR	1		8EPSOIC	40V Low Quiescent Current 150ma Linear Regulator
ISL89411IBZ	U1_DRV	1		SOIC	HS Dual Channel POWER MOSFET Drivers
JUMPER2_100	JP1_BFE1,JP1_BFE2,JP2_BFE1,JP2_BFE2	4		THOLE	Two Pin Jumper
KTR03EZPF10R0	R188_DRV,R191_DRV	2	10	0603	Thick Film Chip Resistor
LMK107B7225KA-T	C63_BFE2,C65_BFE1	2	2.2UF	0603	CERAMIC CAP

MCR03EZPFX1001	R88_BFE1	1	1K	0603	Metal Film Chip Resistor
MCR03EZPFX4701	R3_RA,R4_RA	2	4.7K	0603	Metal Film Chip Resistor
MMBT3904	Q2_PWR	1		SOT23	NPN Transistor
MMBZ5246B	D1_DRV,D3_DRV	2		SM3	16V 350mW Zener Diode
PSK-65A	F1	1	65A	SMD	0.8 mohm Surface Mount Current Protector Fuse
PZTA06	Q15_BFE1,Q15_BFE2	2		SOT223	Vceo 80V 500mA General Purpose NPN Transistor
R7FA2E1A93CFM	U1_RA	1		64TQFP	Ultra Low Power 48MHz Core Up to 128KB Memory and 12-Bit A/D Converter
RAA2230124GNP#JA0	U1_PWR	1		SOT23_5	700V AC/DC Buck Regulator with Ultra-Low Standby Power
RAA489204ANZ	U1_BFE1,U1_BFE2	2		64TQFP	Industrial Multi-Cell Li-Ion Battery Manager
RSR010N10FHA	Q1_DRV	1		SOT23	N-Channel 100V 1A 1 Watt Nch Power MOSFET
S0603CPX4990F10	R22_BFE1,R22_BFE2,R23_BFE1,R23_BFE2,R24_BFE1,R24_BFE2,R25_BFE1,R25_BFE2,R26_BFE1,R26_BFE2,R47_BFE1,R47_BFE2,R48_BFE1,R48_BFE2,R49_BFE1,R49_BFE2,R50_BFE1,R50_BFE2,R51_BFE1,R51_BFE2,R69_BFE1,R70_BFE1,R70_BFE2,R71_BFE1,R71_BFE2,R72_BFE1,R72_BFE2,R73_BFE1,R73_BFE2,R74_BFE2,R83_BFE1,R87_BFE2	32	499	0603	Thick Film Chip Resistor
SMBJ5381B-TP	D2_DRV	1		SM2	130V 5W Zener Diode
SMCJ60A	D3_BFE1,D3_BFE2	2		DO-214	60V 1500 WATT TRANSIENT VOLTAGE SUPPRESSON DIODE
SMD-60C-PAD-DNP	P1_RA-P8_RA,P1_BFE1,P2_BFE1	10		SMD	60 Mil Circular Surface Mount
SQM85N15-19	Q30_DRV,Q31_DRV	2		D2PAK	150V 85A N-CHANNEL (D-S) MOSFET (Automotive - AEC-Q101)
SRR1005-471K	L1_PWR	1	470UH	SMD-S	SMT Shielded Inductor
SSQ-106-02-T-D-RA-MOD	J2_RA	1		CONN2X6	12 Pin 2 X 6 2.54mm x 2.54mm (.100) Right Angle Connector

TMK316B7106KL-TD	C9_PWR,C11_PWR,C135_DRV	3	10UF	1206	MULTILAYER CAP
UMK107AB7105KA	C3_PWR	1	1UF	0603	Multilayer Cap
ZXMP10A13FQTA	Q2	1		SOT23	100V 600mA P-channel MOSFET
C1608X5R1E106M080AC	C8, C4_DRV	2	10uF	0603	CAP CER 10UF 25V X5R 0603
RMCF0603FT499K	R2	1	499K	0603	499k Ohms ±1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thick Film
RMCF0603FT13K0	R1	1	13K	0603	13k Ohms ±1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thick Film
CGA3E3X5R1H105M080AB	C10_PWR	1	1uF	0603	CAP CER 1UF 50V X5R 0603

Appendix A. Software API

The list of functions included in the demo project is shown below, with a brief description of each function. For further information regarding input parameters and return values, refer to the demo project source code.

Auxi Functions

- void **auxi_init** (void)
  Opens the UART driver.

- void **auxi_deinit** (void)
  Closes the UART driver.

- void **auxi_callback** (uart_callback_args_t *p_args)
  Callback for the UART transmission interrupt. If data was received, places data in receive buffer until 'Enter' is received.

- void **auxi_write** (uint8_t *p_msg)
  Performs a UART write with a given input string. Checks the transmission was completed within the timeout limit.

- void **auxi_read** (uint8_t *rec_buf, uint8_t buf_size)
  Reads the UART input buffer and places the contents in the receive buffer.

BMS Functions

- void **bms_init** (void)
  Opens the SPI driver.

- void **bms_deinit** (void)
  Close the SPI driver.

- void **bms_setup** (void)
  Sets up the BMS registers to initialize the BMS.

- void **bms_roll_call** (void)
  Performs a roll call command to establish the daisy chain. Required as the first transmission and after certain reset conditions.

- void **bms_command** (uint8_t dev_addr, uint8_t reg_addr, uint16_t reg_val, uint8_t num_bytes)
  Sends a BMS command to a BMS device in the daisy chain.

- uint16_t **bms_get_reg** (uint8_t dev_addr, uint8_t reg_addr, uint16_t reg_val, uint8_t num_bytes)
  Performs a single register read from a BMS device in the daisy chain.

- void **bms_get_multi_reg** (uint8_t dev_addr, uint8_t reg_addr, uint16_t *reg_val, uint8_t num_bytes)
  Performs a multi-register read from a BMS device in the daisy chain.

- void **bms_set_reg** (uint8_t dev_addr, uint8_t reg_addr, uint16_t reg_val, uint8_t num_bytes)
  Performs a single register write to a BMS device in the daisy chain.

- void **bms_set_multi_reg** (uint8_t dev_addr, uint8_t reg_addr, uint16_t *reg_val, uint8_t num_bytes)
  Performs a multi-register write to a BMS device in the daisy chain.

- void **bms_spi_callback** (spi_callback_args_t *p_args)
  Callback for the BMS SPI interrupt.

- uint16_t **bms_crc16_calculate** (uint32_t numbytes, uint8_t *input_buf)
  Calculates the CRC-16 value for a given input buffer.

- uint32_t **bms_crc32_calculate** (uint32_t numbytes, uint32_t *input_buf)
  Calculates the CRC-32 value for a given input buffer.

- void **bms_scan_voltages** (void)
  Scans the cell/pack voltages for all of the BMS devices in the daisy chain.

- void **bms_scan_temps** (void)
  Scans the pack voltage, internal and external temperatures for all of the BMS devices in the daisy chain.

- void **bms_scan_mixed** (void)
  Scans the cell/pack voltages and external temperature 1 for all of the BMS devices in the daisy chain.

- void **bms_scan_wires** (void)
  Scans all of the BMS devices in the daisy chain for open wire inputs.

- void **bms_scan_all** (void)
  Scans all of the BMS device properties (voltages, temperatures, wires, etc.)

- void **bms_scan_continuous** (void)
  Commands each device in the daisy chain stack to switch to continuous scan mode.

- void **bms_scan_inhibit** (void)
  Stops the Scan Continuous function.

- void **bms_measure** (uint8_t device, uint8_t length)
  Commands the target BMS device to measure a single cell/pack voltage, temperature, etc. The length input value defines which signal is measured (refer to datasheet)

- void **bms_sleep** (void)
  Signals to all devices in the daisy chain to go to sleep.

- void **bms_wakeup** (void)
  Signals to all devices in the daisy chain to wake up from sleep mode.

- void **bms_soft_reset** (void)

Performs a software reset on each device in the daisy chain stack.

- void **bms_hard_reset** (void)
  Performs a hardware reset on each device in the daisy chain stack.

- float **bms_get_cell** (uint8_t device, uint8_t cell)
  Reads the signed digital value for a given cell or pack voltage Converts the value to a signed cell or pack voltage.

- void **bms_get_all_cells** (uint8_t device)
  Reads the digital value for the 4 external temperatures connected to the BMS. Converts the values to temperature values in degrees Celsius.

- float **bms_get_temp** (uint8_t device)
  Reads the digital value for the internal temperature of the BMS. Converts the value to a temperature value in degrees Celsius.

- void **bms_get_external_temps** (uint8_t device)
  Reads the digital value for the 4 external temperatures connected to the BMS. Converts the values to temperature values in degrees Celsius.

- float **bms_get_vref** (uint8_t device)
  Reads the digital value for the reference voltage of the BMS. Converts the value to a voltage.

- uint16_t **bms_get_fault_status** (uint8_t device)
  Reads the fault status register of the BMS.

- uint16_t **bms_get_ov_fault** (uint8_t device)
  Reads the overvoltage fault status register of the BMS.

- uint16_t **bms_get_uv_fault** (uint8_t device)
  Reads the undervoltage fault status register of the BMS.

- uint16_t **bms_get_ow_fault** (uint8_t device)
  Reads the open wire fault status register of the BMS.

- uint16_t **bms_get_ot_fault** (uint8_t device)
  Reads the overtemperature fault status register of the BMS.

- void **bms_fault_setup** (uint8_t device, uint16_t reg_val)
  Sets the fault setup register (external temperatures, fault totalizer, etc.)

- void **bms_internal_temp_warning** (uint8_t device, uint16_t reg_val)
  Sets the internal temperature warning level in the BMS.

- void **bms_internal_temp_limit** (uint8_t device, uint16_t reg_val)
  Sets the internal temperature limit in the BMS.

- void **bms_cell_balance** (uint8_t device)
  Performs cell balancing after disabling balancing, taking readings, and confirming there are no faults. The minimum cell value is compared with all other cells. If the cell voltage difference exceeds a given delta, balancing is enabled for that cell.

- void **bms_clear_all_faults** (uint8_t device)
  Clears all faults in the device by clearing the fault registers and fault status register.

- void **bms_update** (void)
  Updates the BMS readings and checks for BMS faults. Performs cell balancing if there is no load connected.

DPM Functions

- void **dpm_init** (void)
  Opens the I2C driver.

- void **dpm_deinit** (void)
  Closes the I2C driver.

- void **dpm_setup** (void)
  Sets up the DPM registers to initialize the DPM.

- void **dpm_i2c_callback** (i2c_master_callback_args_t *p_args)
  Callback for the I2C transmission interrupt.

- void **dpm_get_reg** (uint8_t reg_addr, uint8_t *reg_val, uint8_t num_bytes)
  Reads the value of a DPM register.

- void **dpm_set_reg** (uint8_t reg_addr, uint16_t reg_val, uint8_t num_bytes)
  Sets the value of a DPM register.

- float **dpm_read_vshunt** (void)
  Reads the signed digital value in the DPM auxiliary vshunt register. The value is converted to a voltage.

- float **dpm_read_vout** (void)
  Reads the unsigned digital value in the DPM vout register. The value is converted to a voltage.

- float **dpm_read_iout** (void)
  Reads the signed digital value in the DPM iout register. The value is converted to a current.

- float **dpm_read_vout_aux** (void)
  Reads the unsigned digital value in the DPM auxiliary vbus register. The value is converted to a voltage.

- float **dpm_read_temp** (void)
  Reads the signed digital value in the DPM temperature register. The value is converted to the temperature in degrees Celsius.

- uint8_t **dpm_read_status_vout** (void)
  Reads the fault status of the DPM voltage registers.

- uint16_t **dpm_read_status_iout** (void)
  Reads the fault status of the DPM current register.

- uint16_t **dpm_read_status_temp** (void)
  Reads the fault status of the DPM temperature registers.

- uint16_t **dpm_read_status_word** (void)
  Reads the aggregate fault status of the DPM (VOUT, IOUT, etc.)

- void **dpm_clear_faults** (void)
  Clears all fault readings in the DPM.

- void **dpm_soft_reset** (void)
  Resets the DPM to its default settings.

- void **dpm_set_iout_oc_threshold** (uint8_t iout_dir, uint8_t vshunt_rng, double oc_value)
  Sets the overcurrent threshold and current direction for detection based on user input value.

- void **dpm_update** (void)
  Updates the DPM readings and checks for DPM register faults and overcurrent faults.

System Functions

- void **sys_modules_init** (void)
  Initializes all of the system modules (except for comms such as I2C)

- void **sys_modules_deinit** (void)
  Closes all of the system driver modules (communications, interrupts, etc.)

- void **sys_5cnt_callback** (timer_callback_args_t *p_args)
  Callback for the 5-second timer interrupt.

- void **sys_20cnt_callback** (timer_callback_args_t *p_args)
  Callback for the 20-second timer interrupt.

- void **sys_count_callback** (timer_callback_args_t *p_args)
  Callback for the timer interrupt for turning off FETs if current drops to 0.

- void **sys_icount_callback** (timer_callback_args_t *p_args)
  Callback for the timer interrupt for turning FET back on if battery or current returns to normal levels.

- void **sys_fet_callback** (timer_callback_args_t *p_args)
  Callback for the timer interrupt for turning off FETs if current is not present.

- void **adc_callback** (adc_callback_args_t *p_args)
  Callback for the ADC completion interrupt.

- double **adc_read** (void)
  Reads the ADC results.

- void **irq_master_callback** (external_irq_callback_args_t *p_args)
  Callback for the BMS fault pin interrupt.

- void **irq_imon_callback** (external_irq_callback_args_t *p_args)
  Callback for the DPM overcurrent interrupt.

- void **irq_datar_callback** (external_irq_callback_args_t *p_args)
  Callback for the BMS DATAREADY pin interrupt.

- void **irq_load_callback** (external_irq_callback_args_t *p_args)
  Callback for the load detect interrupt.

- void **irq_chg_callback** (external_irq_callback_args_t *p_args)
  Callback for the charger detect interrupt.

- void **sys_echo** (uint8_t *buf, uint8_t length)
  Prints the contents of the RS-485 read buffer locally and clears it.

- void **sys_pin_write** (bsp_io_port_pin_t pin, uint8_t level)
  Wrapper function for writing to GPIO pins.

- void **sys_cfet_on_off** (uint8_t level)
  Controls the CFET.

- void **sys_dfet_on_off** (uint8_t level)
  Controls the DFET.

- void **sys_fet_driver_on_off** (uint8_t level)
  Controls the FET driver.

- void **sys_init** (void)
  Sets the system to its initial state and initializes all system driver modules.

- void **sys_setup** (void)
  Performs DPM and BMS register initialization and resets all flags.

- void **sys_power_on_reset** (void)
  Checks the system VCC, VOUT, and battery levels on initialization.

- void **sys_ov_uv_check** (void)
  Finds the minimum and maximum cell voltages and compares them to OV and UV limits. If a limit is hit, the corresponding FET is turned off.

- void **sys_fuel_gauge** (void)
  Monitors the battery level of both battery packs and reports the level to the user.

- void **sys_send_diagnostics** (void)
  Periodically sends diagnostics data such as current, vpack level, and fuel gauge.

- void **sys_user_command** (void)
  Checks user input via RS-485 and performs string matching with known command list to execute user commands.

- void **sys_dpm_command** (void)
  Interprets user input and updates DPM registers based on address, value, and length.

- void **sys_shutdown** (void)
  Shuts the system down when major faults are detected.

- void **sys_restart** (void)
  Attempts to restart the system by restarting all drivers, faults, and flags.

- void **sys_fault_handle** (void)
  System fault handler for detecting major faults and shutting down the system.

- void **sys_oc_handle** (void)
  Checks for over-current (with inrush current handling) using the current reading from the DPM.

- void **sys_chg_load_detect** (void)
  Handles load/charger detection and ready state based on if current is present or not.

- void **sys_chg_dischg_procedure** (void)
  Handles charge/discharge procedures by monitoring current levels, checking cell limits, and using timeouts.

- void **sys_get_vpack** (void)
  Checks battery level with limits and compares MCU and BMS readings.

- void **sys_traceback** (uint8_t func, uint8_t err_flag)
  Provides error traceback and system readings when the system goes into shutdown.

- void **sys_main** (void)
  main function for battery management system.

Appendix B. Pin Functions

The tables below outline the pins used in the demo project and their function.

MCU PIN Name	Net Name	Pin	Pin Config	Sub Block	Default	Description
P401	CFET_DFET PWROFF	2	Output (Open Drain)	FET Driver	0	Controls the system 12V power
P402	DFET_CTRL _MCU	3	Output (Push-pull)	FET Driver	1	Logic input to FET driver
P403	CFET_CTRL _MCU	4	Output (Push-pull)	FET Driver	1	Logic input to FET driver

MCU PIN Name	Net Name	Pin	Pin Config	Sub Block	Default	Description
P408	I2C_SCLK	15	Output SCL (Open Drain)	Current Sense	1	I2C Comm pin
P407	I2C_SDA	16	Output SDA (Open Drain)	Current Sense	1	I2C Comm pin
P004	ISENSE_OC	60	Input IRQ03	Current Sense	0	Overcurrent interrupt

MCU PIN Name	Net Name	Pin	Pin Config	Sub Block	Default	Description
P207	DE/RE _MCU	21	Output (Push-pull)	RS-485	0	UART Flow Control
P206	RX_MCU	22	Input RX	RS-485	0	UART RX
P205	TX_MCU	23	Output (Push-pull)	RS-485	1	UART TX

MCU PIN Name	Net Name	Pin	Pin Config	Sub Block	Default	Description
P105	FAULT _MASTER _MCU	43	Input IRQ00	BMS	1	BMS System Fault Interrupt
P104	DATAREADY _MCU	44	Input IRQ01	BMS	1	SPI Data Ready interrupt
P103	CS_MCU	45	Output SSL0 (Push-pull)	BMS	1	SPI Slave Select 0
P102	SPICLK	46	Output RSPCK (Push-pull)	BMS	1	SPI Clock Line
P101	MOSI_MCU	47	Output MOSI (Push-pull)	BMS	1	SPI Master Out Slave In
P100	MISO_MCU	48	Input MISO	BMS	1	SPI Master In Slave Out

MCU PIN Name	Net Name	Pin	Pin Config	Sub Block	Default	Description
P000	VBAT_ADC	64	Input AN00	ADC	1	ADC input for VBAT
P411	LD_DET_MCU	12	Input IRQ04	System	1	Interrupt to detect load connection
P410	CHRG_DET _MCU	11	Input IRQ05	System	1	Interrupt to detect charger connection