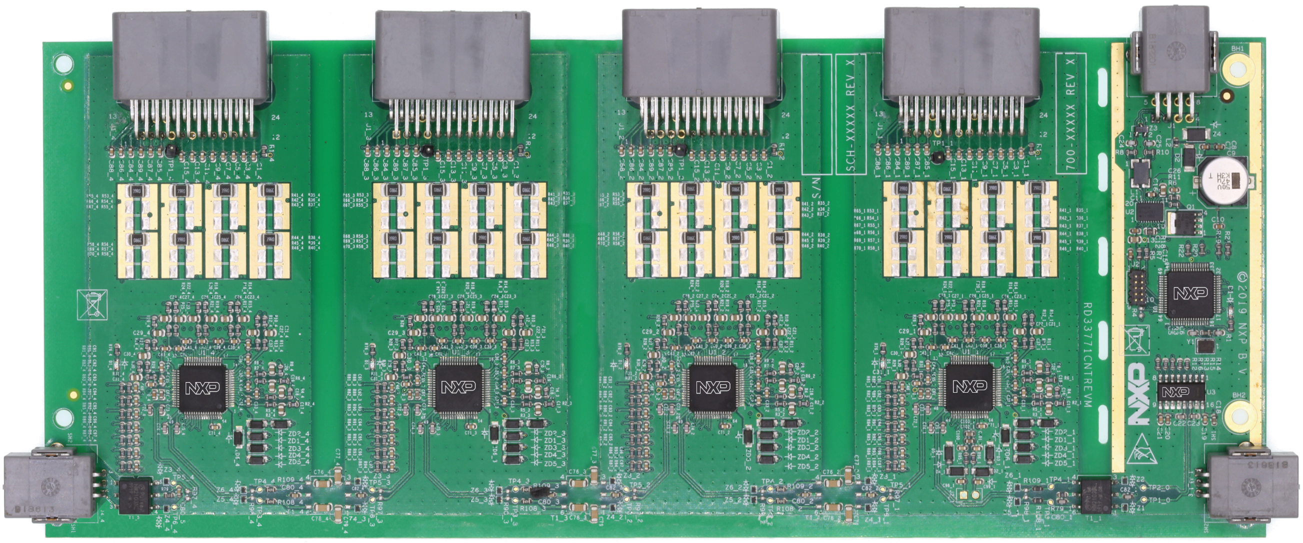
UM11310

HV Battery management system

Rev. 1 — 7 January 2020 User manual



aaa-035350

NXP provides the enclosed product(s) under the following conditions:

This reference design is intended for use of ENGINEERING DEVELOPMENT OR EVALUATION PURPOSES ONLY. It is provided as a sample IC pre-soldered to a printed circuit board to make it easier to access inputs, outputs, and supply terminals. This reference design may be used with any development system or other source of I/O signals by simply connecting it to the host MCU or computer board via off-the-shelf cables. Final device in an application will be heavily dependent on proper printed circuit board layout and heat sinking design as well as attention to supply filtering, transient suppression, and I/O signal quality.

The goods provided may not be complete in terms of required design, marketing, and or manufacturing related protective considerations, including product safety measures typically found in the end product incorporating the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge. In order to minimize risks associated with the customers applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards. For any safety concerns, contact NXP sales and technical support services.

Should this reference design not meet the specifications indicated in the kit, it may be returned within 30 days from the date of delivery and will be replaced by a new kit.

NXP reserves the right to make changes without further notice to any products herein. NXP makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typical”, must be validated for each customer application by customer’s technical experts.

NXP does not convey any license under its patent rights nor the rights of others. NXP products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the NXP product could create a situation where personal injury or death may occur. Should the Buyer purchase or use NXP products for any such unintended or unauthorized application, the Buyer shall indemnify and hold NXP and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges NXP was negligent regarding the design or manufacture of the part.

1 Finding kit resources and information on the NXP web site

NXP Semiconductors provides online resources for this evaluation board and its supported device(s) on http://www.nxp.com.

The information page for RD33771CNTREVM, HV battery management system reference design is at http://www.nxp.com/products/:RD33771CNTREVM. The information page provides overview information, documentation, software and tools, parametrics, ordering information and a Getting Started tab. The Getting Started tab provides quick-reference information applicable to using the RD33771CNTREVM, HV Battery management system reference design, including the downloadable assets referenced in this document.

1.1 Collaborate in the NXP Community

The NXP Community is for sharing ideas and tips, asking and answering technical questions, and receiving input on just about any embedded design topic.

The NXP Community is at http://community.nxp.com.

2 Getting started

2.1 Kit contents

The **RD33771CNTREVM** contents include:

- RD33771CNTREVM reference design board
- Battery simulation cable x 4 – power supply for each BCC, simulate each cell by series resistors.
- Low-voltage cable x 1 – power supply SBC, support CAN communication,
- TPL daisy chain cable x 1 – provide a cable to communicate with other BCC boards by TPL.

2.2 Additional hardware

To use this kit, you need:

- Power supply 12 VDC with current capability 500 mA
- Power supply 10 to 50 VDC with current capability 500 mA or a 7-to-14-cell battery pack
- USB Multilink FX debug probe

2.3 Windows PC workstation

The kit requires the following to function properly with the demo software:

- Windows® 10, Windows XP, Windows 7, or Vista in 32- or 64-bit version.

2.4 Software

Installing software is recommended to work with this evaluation board. All listed software is available on the evaluation board's information page [1].

- S32DS-ARM: S32 Design Studio for Arm

3 Getting to know the hardware

3.1 General description: RD33771CNTREVM

The RD33771CNTREVM provides a solution for a centralized and distributed architecture for lithium-ion battery management in automotive applications.

This board allocates four MC33771C devices controlled by one MCU. The MCU could be bypassed and stacked to a long daisy chain for a flexible BMS architecture. Each BCC can measure lithium batteries having 7 to 14 cells each. The BCCs communicate by TPL daisy chain or capacitor Isolation. The MCU is supplied by an SBC that is powered by a 12 VDC power source.

3.2 Features: RD33771CNTREVM

- Four BCCs on one board
- Each BCC can measure voltage for up to 14 battery cells with high accuracy
- Each BCC has 6-channel temperature sensing
- The first BCC has a current-sensing point, could connect an external SHUNT resistor for current measurement
- Capacitor or TPL isolation communication on the board
- TPL isolation for off-board communication
- Cell balancing current set to 100 mA, expandable up to 300 mA by adding CB resistors
- Low cost SBC and MCU as BCC management
- One channel CAN interface
- JTAG debugging interface
- High EMC performance: passed BCI 200 mA, CE CISPR class 3

3.3 Board functions

This reference design board provide design example for following functions:

Index	Function	Description
1	MCU communicates with BCC by high-voltage capacitor	• MCU software communicates with first BCC by transformer, others by high-voltage capacitor, baud rate at 2 Mbps
2	TPL Isolation communication	• MCU software communicates with 4 BCCs on the board by transformer, baud rate at 2 Mbps
3	100 to 300 mA cell balancing heating	• Each BCC channel has 100 to 300 mA cell balancing capability, change/add cell balancing resistors adjusts current
4	BCC diagnostic polling to CAN	• MCU software cyclic run diagnostic mechanism and upload result to CAN
5	BCC ADC acquisition polling to CAN	• MCU software cyclic launch conversion of BCC ADC and upload cell voltage/temperature/current (optional) to CAN
6	EMC performance validated	• BCI 200 mA open loop max accuracy error < 6 mV, high voltage cell terminals harness length 2 m, TPL cable length 2 m • CE pass CISPR class3 • For more details, refer to the EMC validation chapter.

3.4 Block diagram

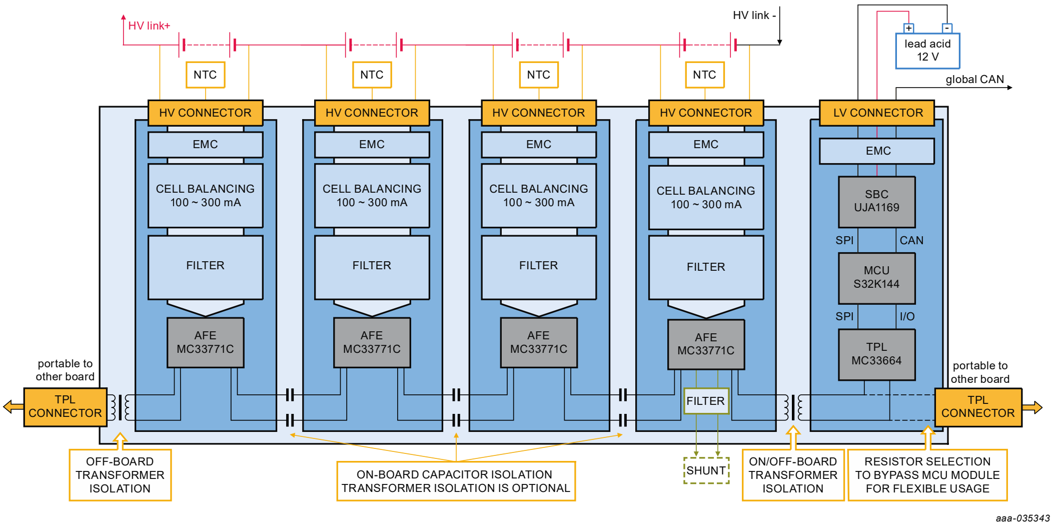


Figure 1. Block diagram

3.5 Reference design featured components

The RD33771CNTREVM allows the user to exercise all the functions of the MC33771C.

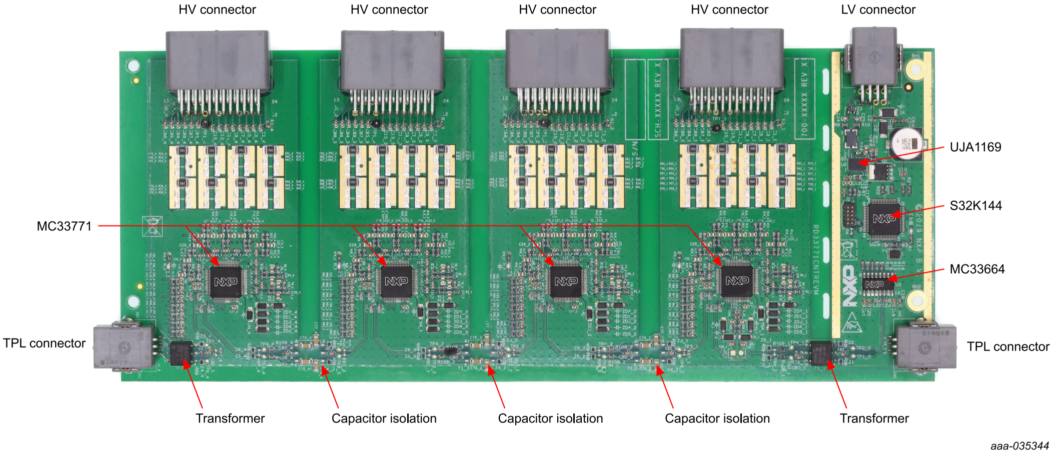


Figure 2. Board description

Table 1. Board description

Name	Description
MC33771C	14-Channel Li-ion battery cell controller IC
MC33664	Isolated communication IC
UJA1169	Mini high-speed CAN companion system basis chip
S32K144	32-bit automotive general purpose microcontroller

3.5.1 Devices and features

This reference design/evaluation board features the following NXP products:

Table 2. Device features

Device	Description	Features
MC33771C	Battery cell controller; 14-channel analog front end (AFE)	• 9.6 V ≤ V_{PWR} ≤ 61.6 V operation, 75 V transient • 7 to 14 cells management • Isolated 2.0 Mbps differential communication or 4.0 Mbps SPI • Addressable on initialization • Bidirectional transceiver to support up to 63 nodes in daisy chain • 0.8 mV maximum total voltage measurement error • Synchronized cell voltage/current measurement with coulomb count • Averaging of cell voltage measurements • Total stack voltage measurement • Seven GPIO/temperature sensor inputs • 5.0 V at 5.0 mA reference supply output • Automatic over/undervoltage and temperature detection routable to fault pin • Integrated sleep mode over/undervoltage and temperature monitoring • Onboard 300 mA passive cell balancing with diagnostics • Hot plug capable • Detection of internal and external faults, as open lines, shorts, and leakages • Designed to support ISO 26262, up to ASIL D safety capability • Qualified in compliance with AECQ-100
MC33664	Isolated network high-speed transceiver; transformer physical layer (TPL)	• 2.0 Mbit/s isolated network communication rate • Dual SPI architecture for message confirmation • Robust conducted and radiated immunity with wake-up • 3.3 V and 5.0 V compatible logic thresholds • Low sleep mode current with automatic bus wake-up • Ultra-low radiated emissions
UJA1169	Mini high-speed CAN companion system basis chip	• ISO 11898-2:201x (upcoming merged ISO 11898-2/5/6) compliant 1 Mbit/s high-speed CAN transceiver supporting CAN FD active communication up to 2 Mbit/s in the CAN FD data field (all six variants) • Autonomous bus biasing according to ISO 11898-6:2013 and ISO 11898-2:201x • Scalable 5 V or 3.3 V 250 mA low-drop voltage regulator for 5 V/3.3 V microcontroller supply (V1) based on external PNP transistor concept for thermal scaling • CAN-bus connections are truly floating when power to pin BAT is off • No ‘false’ wake-ups due to CAN FD traffic (in variants supporting partial networking)
S32K144	32-bit Automotive General Purpose Microcontrollers	• Operating characteristics – Voltage range: 2.7 V to 5.5 V – Ambient temperature range: -40 °C to 105 °C for HSRUN mode, -40 °C to 125 °C for RUN mode • Arm™ Cortex-M4F/M0+ core, 32-bit CPU – Supports up to 112 MHz frequency (HSRUN mode) with 1.25 Dhrystone MIPS per MHz – Arm Core based on the Armv7 Architecture and Thumb®-2 ISA – Integrated Digital Signal Processor (DSP) – Configurable Nested Vectored Interrupt Controller (NVIC) – Single Precision Floating Point Unit (FPU)

- Clock interfaces
  - 4 - 40 MHz fast external oscillator (SOSC) with up to 50 MHz DC external square input clock in external clock mode
  - 48 MHz Fast Internal RC oscillator (FIRC)
  - 8 MHz Slow Internal RC oscillator (SIRC)
  - 128 kHz Low Power Oscillator (LPO)
  - Up to 112 MHz (HSRUN) System Phased Lock Loop (SPLL)
  - Up to 20 MHz TCLK and 25 MHz SWD_CLK
  - 32 kHz Real Time Counter external clock (RTC_CLKIN)
- Power management
  - Low-power Arm Cortex-M4F/M0+ core with excellent energy efficiency
  - Power Management Controller (PMC) with multiple power modes: HSRUN, RUN, STOP, VLPR, and VLPS. Note: CSEc (Security) or EEPROM writes/erase will trigger error flags in HSRUN mode (112 MHz) because this use case is not allowed to execute simultaneously. The device will need to switch to RUN mode (80 Mhz) to execute CSEc (Security) or EEPROM writes/erase.
  - Clock gating and low power operation supported on specific peripherals.
- Memory and memory interfaces
  - Up to 2 MB program flash memory with ECC
  - 64 KB FlexNVM for data flash memory with ECC and EEPROM emulation. Note: CSEc (Security) or EEPROM writes/erase will trigger error flags in HSRUN mode (112 MHz) because this use case is not allowed to execute simultaneously. The device will need to switch to RUN mode (80 MHz) to execute CSEc (Security) or EEPROM writes/erase.
  - Up to 256 KB SRAM with ECC
  - Up to 4 KB of FlexRAM for use as SRAM or EEPROM emulation
  - Up to 4 KB Code cache to minimize performance impact of memory access latencies
  - QuadSPI with HyperBus™ support
- Mixed-signal analog
  - Up to two 12-bit Analog-to-Digital Converter (ADC) with up to 32 channel analog inputs per module
  - One Analog Comparator (CMP) with internal 8-bit Digital to Analog Converter (DAC)
- Debug functionality
  - Serial Wire JTAG Debug Port (SWJ-DP) combines
  - Debug Watchpoint and Trace (DWT)
  - Instrumentation Trace Macrocell (ITM)
  - Test Port Interface Unit (TPIU)
  - Flash Patch and Breakpoint (FPB) Unit
- Human-machine interface (HMI)
  - Up to 156 GPIO pins with interrupt functionality
  - Non-Maskable Interrupt (NMI)
- Communications interfaces
  - Up to three Low Power Universal Asynchronous Receiver/Transmitter (LPUART/LIN) modules with DMA support and low power availability
  - Up to three Low Power Serial Peripheral Interface (LPSPI) modules with DMA support and low power availability
  - Up to two Low Power Inter-Integrated Circuit (LPI2C) modules with DMA support and low power availability
  - Up to three FlexCAN modules (with optional CAN-FD support)
  - FlexIO module for emulation of communication protocols and peripherals (UART, I2C, SPI, I2S, LIN, PWM, etc).
  - Up to one 10/100Mbps Ethernet with IEEE1588 support and two Synchronous Audio Interface (SAI) modules.

		• Safety and Security – Cryptographic Services Engine (CSEc) implements a comprehensive set of cryptographic functions as described in the SHE (Secure Hardware Extension) Functional Specification. Note: CSEc (Security) or EEPROM writes/erase will trigger error flags in HSRUN mode (112 MHz) because this use case is not allowed to execute simultaneously. The device will need to switch to RUN mode (80 MHz) to execute CSEc (Security) or EEPROM writes/erase. – 128-bit Unique Identification (ID) number – Error-Correcting Code (ECC) on flash and SRAM memories – System Memory Protection Unit (System MPU) – Cyclic Redundancy Check (CRC) module – Internal watchdog (WDOG) – External Watchdog monitor (EWM) module • Timing and control – Up to eight independent 16-bit FlexTimers (FTM) modules, offering up to 64 standard channels (IC/OC/PWM) – One 16-bit Low Power Timer (LPTMR) with flexible wake up control – Two Programmable Delay Blocks (PDB) with flexible trigger system – One 32-bit Low Power Interrupt Timer (LPIT) with 4 channels – 32-bit Real Time Counter (RTC) • Package – 32-pin QFN, 48-pin LQFP, 64-pin LQFP, 100-pin LQFP, 100-pin MAPBGA, 144-pin LQFP, 176-pin LQFP package options • 16 channel DMA with up to 63 request sources using DMAMUX

3.6 Connectors

Figure 3 shows the location of connectors on the board. The tables in this section list the pinouts for each connector.

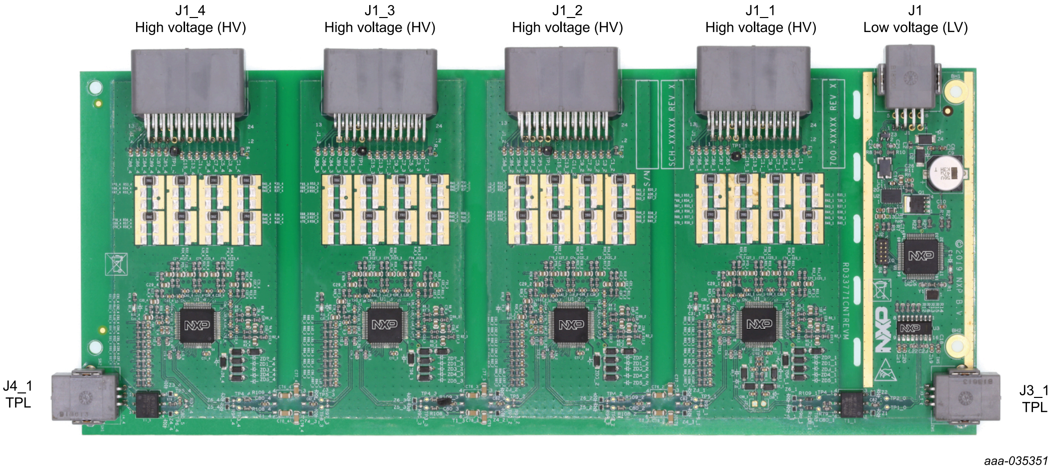


Figure 3. Connectors

Table 3. J1, Low voltage connector

Pin	Name	Description
1	CAN0_L	CAN bus negative
2	CAN0_H	CAN bus positive
3	VBAT	Power supply for MCU, 12VDC
4	VBAT	Power supply for MCU, 12VDC

Pin	Name	Description
5	GND	Ground of MCU
6	GND	Ground of MCU
7	GND	Ground of MCU
8	GND	Ground of MCU

Table 4. J1_1, High voltage connector for BCC1

Pin	Name	Description
1	B1_NTC_GPIO5	Should connect to external NTC resistor for temperature sensing
2	B1_NTC_GPIO3	Should connect to external NTC resistor for temperature sensing
3	B1_NTC_GPIO1	Should connect to external NTC resistor for temperature sensing
4	GND_B1	Ground of NTC resistors
5	B1_VB_CT_1	Cell1 voltage sensing point
6	B1_VB_CT_3	Cell3 voltage sensing point
7	B1_VB_CT_5	Cell5 voltage sensing point
8	B1_VB_CT_7	Cell7 voltage sensing point
9	B1_VB_CT_9	Cell9 voltage sensing point
10	B1_VB_CT_11	Cell11 voltage sensing point
11	B1_VB_CT_13	Cell13 voltage sensing point
12	VBAT_B1	Power supply for BCC
13	B1_NTC_GPIO4	Should connect to external NTC resistor for temperature sensing
14	B1_NTC_GPIO2	Should connect to external NTC resistor for temperature sensing
15	B1_NTC_GPIO0	Should connect to external NTC resistor for temperature sensing
16	GND_B1	Ground of BCC
17	B1_VB_CT_REF	Cell0 voltage sensing point, Wire should be connected same point with GND_B1
18	B1_VB_CT_2	Cell2 voltage sensing point
19	B1_VB_CT_4	Cell4 voltage sensing point
20	B1_VB_CT_6	Cell6 voltage sensing point
21	B1_VB_CT_8	Cell8 voltage sensing point
22	B1_VB_CT_10	Cell10 voltage sensing point
23	B1_VB_CT_12	Cell12 voltage sensing point
24	B1_VB_CT_14	Cell14 voltage sensing point Wire should be connected at the same point with VBAT_B1

Table 5. J1_2, High Voltage connector for BCC2

Pin	Name	Description
1	B2_NTC_GPIO5	Should connect to external NTC resistor for temperature sensing
2	B2_NTC_GPIO3	Should connect to external NTC resistor for temperature sensing
3	B2_NTC_GPIO1	Should connect to external NTC resistor for temperature sensing
4	GND_B2	Ground of NTC resistors
5	B2_VB_CT_1	Cell1 voltage sensing point
6	B2_VB_CT_3	Cell3 voltage sensing point
7	B2_VB_CT_5	Cell5 voltage sensing point
8	B2_VB_CT_7	Cell7 voltage sensing point
9	B2_VB_CT_9	Cell9 voltage sensing point
10	B2_VB_CT_11	Cell11 voltage sensing point
11	B2_VB_CT_13	Cell13 voltage sensing point
12	VBAT_B2	Power supply for BCC
13	B2_NTC_GPIO4	Should connect to external NTC resistor for temperature sensing
14	B2_NTC_GPIO2	Should connect to external NTC resistor for temperature sensing
15	B2_NTC_GPIO0	Should connect to external NTC resistor for temperature sensing
16	GND_B2	Ground of BCC
17	B2_VB_CT_REF	Cell0 voltage sensing point, Wire should be connected at the same point with GND_B1
18	B2_VB_CT_2	Cell2 voltage sensing point
19	B2_VB_CT_4	Cell4 voltage sensing point
20	B2_VB_CT_6	Cell6 voltage sensing point
21	B2_VB_CT_8	Cell8 voltage sensing point
22	B2_VB_CT_10	Cell10 voltage sensing point
23	B2_VB_CT_12	Cell12 voltage sensing point
24	B2_VB_CT_14	Cell14 voltage sensing point Wire should be connected at the same point with VBAT_B1

Table 6. J1_3, High voltage connector for BCC3

Pin	Name	Description
1	B3_NTC_GPIO5	Should connect to external NTC resistor for temperature sensing
2	B3_NTC_GPIO3	Should connect to external NTC resistor for temperature sensing
3	B3_NTC_GPIO1	Should connect to external NTC resistor for temperature sensing
4	GND_B3	Ground of NTC resistors
5	B3_VB_CT_1	Cell1 voltage sensing point
6	B3_VB_CT_3	Cell3 voltage sensing point
7	B3_VB_CT_5	Cell5 voltage sensing point

Pin	Name	Description
8	B3_VB_CT_7	Cell7 voltage sensing point
9	B3_VB_CT_9	Cell9 voltage sensing point
10	B3_VB_CT_11	Cell11 voltage sensing point
11	B3_VB_CT_13	Cell13 voltage sensing point
12	VBAT_B3	Power supply for BCC
13	B3_NTC_GPIO4	Should connect to external NTC resistor for temperature sensing
14	B3_NTC_GPIO2	Should connect to external NTC resistor for temperature sensing
15	B3_NTC_GPIO0	Should connect to external NTC resistor for temperature sensing
16	GND_B3	Ground of BCC
17	B3_VB_CT_REF	Cell0 voltage sensing point, Wire should be connected same point with GND_B1
18	B3_VB_CT_2	Cell2 voltage sensing point
19	B3_VB_CT_4	Cell4 voltage sensing point
20	B3_VB_CT_6	Cell6 voltage sensing point
21	B3_VB_CT_8	Cell8 voltage sensing point
22	B3_VB_CT_10	Cell10 voltage sensing point
23	B3_VB_CT_12	Cell12 voltage sensing point
24	B3_VB_CT_14	Cell14 voltage sensing point Wire should be connected same point with VBAT_B1

Table 7. J1_4, High Voltage connector for BCC4

Pin	Name	Description
1	B4_NTC_GPIO5	Should connect to external NTC resistor for temperature sensing
2	B4_NTC_GPIO3	Should connect to external NTC resistor for temperature sensing
3	B4_NTC_GPIO1	Should connect to external NTC resistor for temperature sensing
4	GND_B4	Ground of NTC resistors
5	B4_VB_CT_1	Cell1 voltage sensing point
6	B4_VB_CT_3	Cell3 voltage sensing point
7	B4_VB_CT_5	Cell5 voltage sensing point
8	B4_VB_CT_7	Cell7 voltage sensing point
9	B4_VB_CT_9	Cell9 voltage sensing point
10	B4_VB_CT_11	Cell11 voltage sensing point
11	B4_VB_CT_13	Cell13 voltage sensing point
12	VBAT_B4	Power supply for BCC
13	B4_NTC_GPIO4	Should connect to external NTC resistor for temperature sensing
14	B4_NTC_GPIO2	Should connect to external NTC resistor for temperature sensing
15	B4_NTC_GPIO0	Should connect to external NTC resistor for temperature sensing

Pin	Name	Description
16	GND_B4	Ground of BCC
17	B4_VB_CT_REF	Cell0 voltage sensing point, Wire should be connected same point with GND_B1
18	B4_VB_CT_2	Cell2 voltage sensing point
19	B4_VB_CT_4	Cell4 voltage sensing point
20	B4_VB_CT_6	Cell6 voltage sensing point
21	B4_VB_CT_8	Cell8 voltage sensing point
22	B4_VB_CT_10	Cell10 voltage sensing point
23	B4_VB_CT_12	Cell12 voltage sensing point
24	B4_VB_CT_14	Cell14 voltage sensing point Wire should be connected same point with VBAT_B1

Table 8. J2, JTAG

Pin	Name	Description
1	5 V	Power supply for JTAG debugging tool
2	JTAG_TMS	JTAG mode selection
3	GND	Ground
4	JTAG_TCLK	JTAG clock
5	GND	Ground
6	JTAG_TDO	JTAG data out
7	NC	No Connect
8	JTAG_TDI	JTAG data in
9	NC	No Connect
10	JTAG_TRST	JTAG reset

Table 9. J3_1, TPL connector

Pin	Name	Description
Pin	Name	Description
1	RD-TX_P	Positive of TPL twisted wire
2	NC	No Connect
3	RD-TX_N	Negative of TPL twisted wire

Table 10. J4_1, TPL connector

Pin	Name	Description
1	RD-TX_P	Positive of TPL twisted wire
2	NC	No Connect

Pin	Name	Description
3	RD-TX_N	Negative of TPL twisted wire

3.7 Schematic, board layout and bill of materials

The schematic, board layout and bill of materials for the RD33771CNTREVM, HV battery management system reference design are available on the RD33771CNTREVM tool summary page [1].

This centralized board is designed for low-cost application and was not designed to support applications that have functional safety requirements. Therefore, the device selection requirements are :

- The MCU has at least one channel CAN, two channels SPI, several GPIOs for IO control/detection, and enough internal flash/RAM resources for up to 4 x BCC data management.
- The SBC system management chip contains a watchdog, an SPI to communicate with the MCU, and at least one channel CAN physical layer.
- The BCC is designed to measure 7-to-14 cell lithium battery voltages, current and temperature.

3.7.1 BCC Schematic

- The schematic of a single MC33771 is equivalent to the schematic in the latest datasheet
- The isolated communication between BCC is achieved by a high-voltage capacitor
- The isolated communication between BCC and the MCU is achieved by transformer
- There are two TPL connectors that enable connection to other BCC board(s). The MCU module can be bypassed by a jumper.

3.7.2 SBC Schematic

- The MCU supplied by SBC (UJA1169) provides up to two channels of 5 VDC output. One channel for MCU and MC33664 power supply is enough.
- One channel of CAN at the physical layer is used to connect this board to vehicle CAN bus or other Battery Management Unit (BMU).
- One SPI channel that communicates with the MCU manages the SBC registers, feed watchdog, fault detection, etc.

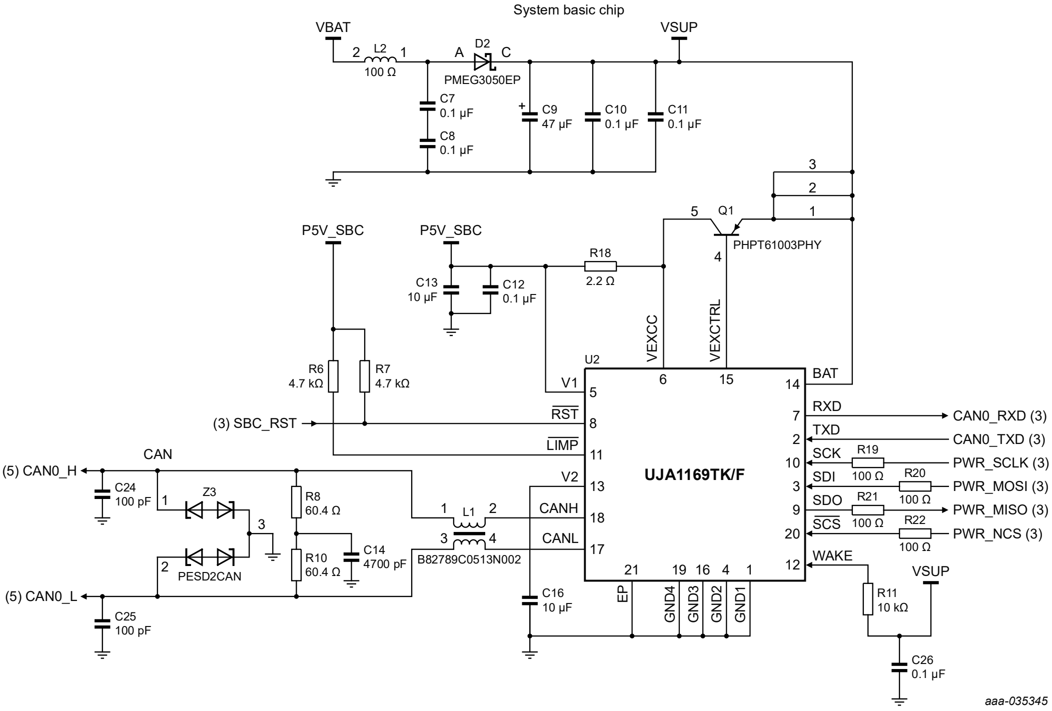


Figure 4. SBC

3.7.3 MCU Schematic

The MCU functions include:

- 1 SPI port to communicate with SBC for power management
- 2 SPI ports connected to the MC33664 for communicating with MC33771s
- 1 CAN port to upload data to the upper layer
- 1 LED as a software flow indicator (debug purpose only)
- 1 JTAG port for program download and debug

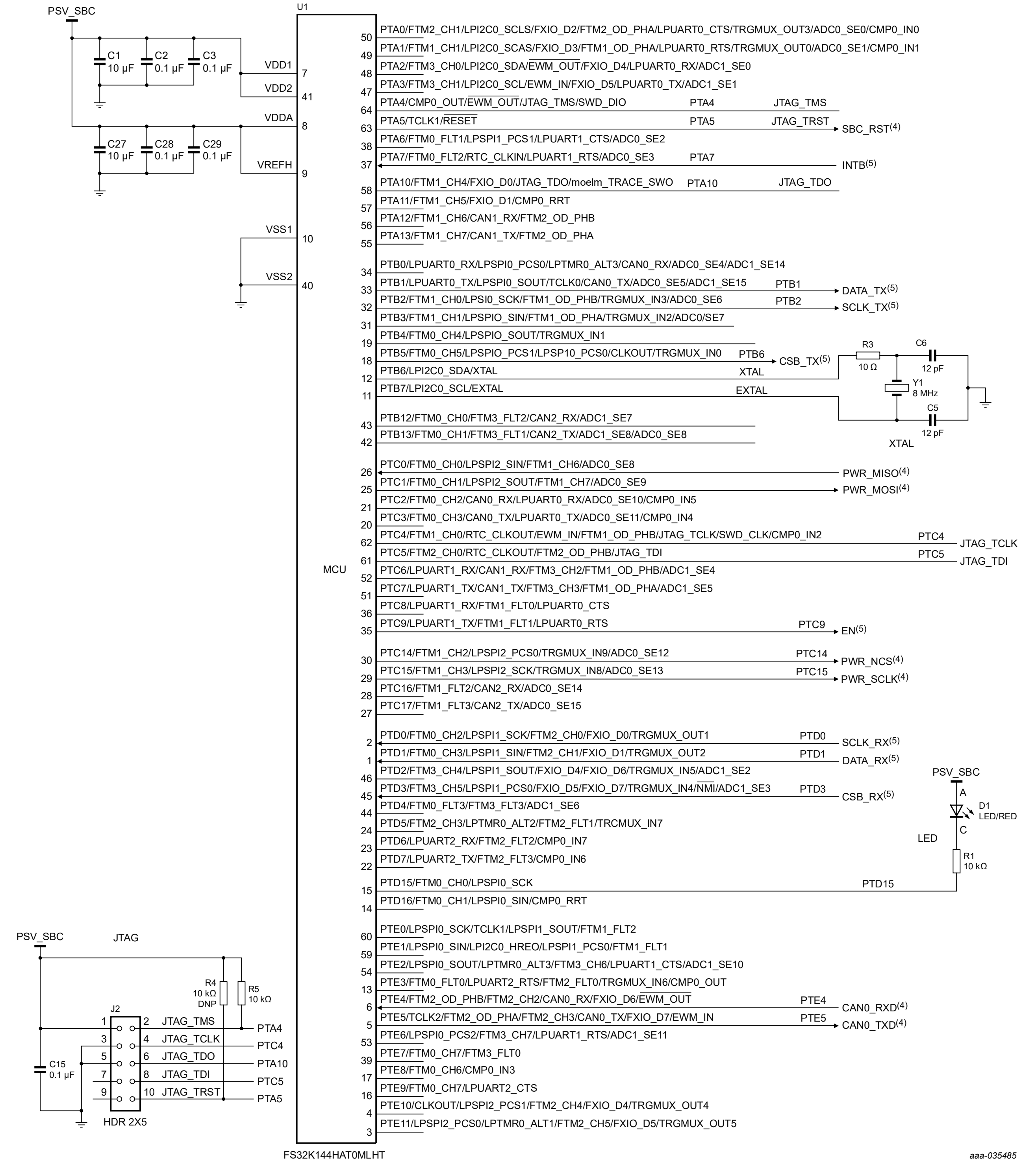

PSV_SBC
C1 10 µF
C2 0.1 µF
C3 0.1 µF
C27 10 µF
C28 0.1 µF
C29 0.1 µF
U1
VDD1 7
VDD2 41
VDDA 8
VREFH 9
VSS1 10
VSS2 40
MCU
50 PTA0/FTM2_CH1/LPI2C0_SCLS/FXIO_D2/FTM2_OD_PHA/LPUART0_CTS/TRGMUX_OUT3/ADC0_SE0/CMP0_IN0
49 PTA1/FTM1_CH1/LPI2C0_SCAS/FXIO_D3/FTM1_OD_PHA/LPUART0_RTS/TRGMUX_OUT0/ADC0_SE1/CMP0_IN1
48 PTA2/FTM3_CH0/LPI2C0_SDA/EWM_OUT/FXIO_D4/LPUART0_RX/ADC1_SE0
47 PTA3/FTM3_CH1/LPI2C0_SCL/EWM_IN/FXIO_D5/LPUART0_TX/ADC1_SE1
64 PTA4/CMP0_OUT/EWM_OUT/JTAG_TMS/SWD_DIO PTA4 JTAG_TMS
63 PTA5/TCLK1/RESET PTA5 JTAG_TRST SBC_RST(4)
38 PTA6/FTM0_FLT1/LPSPI1_PCS1/LPUART1_CTS/ADC0_SE2
37 PTA7/FTM0_FLT2/RTC_CLKIN/LPUART1_RTS/ADC0_SE3 PTA7 INTB(5)
58 PTA10/FTM1_CH4/FXIO_D0/JTAG_TDO/moelm_TRACE_SWO PTA10 JTAG_TDO
57 PTA11/FTM1_CH5/FXIO_D1/CMP0_RRT
56 PTA12/FTM1_CH6/CAN1_RX/FTM2_OD_PHB
55 PTA13/FTM1_CH7/CAN1_TX/FTM2_OD_PHA
34 PTB0/LPUART0_RX/LPSPI0_PCS0/LPTMR0_ALT3/CAN0_RX/ADC0_SE4/ADC1_SE14
33 PTB1/LPUART0_TX/LPSPI0_SOUT/TCLK0/CAN0_TX/ADC0_SE5/ADC1_SE15 PTB1 DATA_TX(5)
32 PTB2/FTM1_CH0/LPSI0_SCK/FTM1_OD_PHB/TRGMUX_IN3/ADC0_SE6 PTB2 SCLK_TX(5)
31 PTB3/FTM1_CH1/LPSPIO_SIN/FTM1_OD_PHA/TRGMUX_IN2/ADC0/SE7
19 PTB4/FTM0_CH4/LPSPIO_SOUT/TRGMUX_IN1
18 PTB5/FTM0_CH5/LPSPIO_PCS1/LPSP10_PCS0/CLKOUT/TRGMUX_IN0 PTB6 CSB_TX(5)
12 PTB6/LPI2C0_SDA/XTAL XTAL
11 PTB7/LPI2C0_SCL/EXTAL EXTAL
R3 10 Ω
C6 12 pF
Y1 8 MHz
C5 12 pF
XTAL
43 PTB12/FTM0_CH0/FTM3_FLT2/CAN2_RX/ADC1_SE7
42 PTB13/FTM0_CH1/FTM3_FLT1/CAN2_TX/ADC1_SE8/ADC0_SE8
26 PTC0/FTM0_CH0/LPSPI2_SIN/FTM1_CH6/ADC0_SE8 PWR_MISO(4)
25 PTC1/FTM0_CH1/LPSPI2_SOUT/FTM1_CH7/ADC0_SE9 PWR_MOSI(4)
21 PTC2/FTM0_CH2/CAN0_RX/LPUART0_RX/ADC0_SE10/CMP0_IN5
20 PTC3/FTM0_CH3/CAN0_TX/LPUART0_TX/ADC0_SE11/CMP0_IN4
62 PTC4/FTM1_CH0/RTC_CLKOUT/EWM_IN/FTM1_OD_PHB/JTAG_TCLK/SWD_CLK/CMP0_IN2 PTC4 JTAG_TCLK
61 PTC5/FTM2_CH0/RTC_CLKOUT/FTM2_OD_PHB/JTAG_TDI PTC5 JTAG_TDI
52 PTC6/LPUART1_RX/CAN1_RX/FTM3_CH2/FTM1_OD_PHB/ADC1_SE4
51 PTC7/LPUART1_TX/CAN1_TX/FTM3_CH3/FTM1_OD_PHA/ADC1_SE5
36 PTC8/LPUART1_RX/FTM1_FLT0/LPUART0_CTS
35 PTC9/LPUART1_TX/FTM1_FLT1/LPUART0_RTS PTC9 EN(5)
30 PTC14/FTM1_CH2/LPSPI2_PCS0/TRGMUX_IN9/ADC0_SE12 PTC14 PWR_NCS(4)
29 PTC15/FTM1_CH3/LPSPI2_SCK/TRGMUX_IN8/ADC0_SE13 PTC15 PWR_SCLK(4)
28 PTC16/FTM1_FLT2/CAN2_RX/ADC0_SE14
27 PTC17/FTM1_FLT3/CAN2_TX/ADC0_SE15
2 PTD0/FTM0_CH2/LPSPI1_SCK/FTM2_CH0/FXIO_D0/TRGMUX_OUT1 PTD0 SCLK_RX(5)
1 PTD1/FTM0_CH3/LPSPI1_SIN/FTM2_CH1/FXIO_D1/TRGMUX_OUT2 PTD1 DATA_RX(5)
46 PTD2/FTM3_CH4/LPSPI1_SOUT/FXIO_D4/FXIO_D6/TRGMUX_IN5/ADC1_SE2
45 PTD3/FTM3_CH5/LPSPI1_PCS0/FXIO_D5/FXIO_D7/TRGMUX_IN4/NMI/ADC1_SE3 PTD3 CSB_RX(5)
44 PTD4/FTM0_FLT3/FTM3_FLT3/ADC1_SE6
24 PTD5/FTM2_CH3/LPTMR0_ALT2/FTM2_FLT1/TRCMUX_IN7
23 PTD6/LPUART2_RX/FTM2_FLT2/CMP0_IN7
22 PTD7/LPUART2_TX/FTM2_FLT3/CMP0_IN6
15 PTD15/FTM0_CH0/LPSPI0_SCK PTD15
14 PTD16/FTM0_CH1/LPSPI0_SIN/CMP0_RRT
PSV_SBC
A
D1 LED/RED
C
LED
R1 10 kΩ
60 PTE0/LPSPI0_SCK/TCLK1/LPSPI1_SOUT/FTM1_FLT2
59 PTE1/LPSPI0_SIN/LPI2C0_HREO/LPSPI1_PCS0/FTM1_FLT1
54 PTE2/LPSPI0_SOUT/LPTMR0_ALT3/FTM3_CH6/LPUART1_CTS/ADC1_SE10
13 PTE3/FTM0_FLT0/LPUART2_RTS/FTM2_FLT0/TRGMUX_IN6/CMP0_OUT
6 PTE4/FTM0_OD_PHB/FTM2_CH2/CAN0_RX/FXIO_D6/EWM_OUT PTE4 CAN0_RXD(4)
5 PTE5/TCLK2/FTM2_OD_PHA/FTM2_CH3/CAN0_TX/FXIO_D7/EWM_IN PTE5 CAN0_TXD(4)
53 PTE6/LPSPI0_PCS2/FTM3_CH7/LPUART1_RTS/ADC1_SE11
39 PTE7/FTM0_CH7/FTM3_FLT0
17 PTE8/FTM0_CH6/CMP0_IN3
16 PTE9/FTM0_CH7/LPUART2_CTS
4 PTE10/CLKOUT/LPSPI2_PCS1/FTM2_CH4/FXIO_D4/TRGMUX_OUT4
3 PTE11/LPSPI2_PCS0/LPTMR0_ALT1/FTM2_CH5/FXIO_D5/TRGMUX_OUT5
FS32K144HAT0MLHT
PSV_SBC
JTAG
R4 10 kΩ DNP
R5 10 kΩ
J2
1 2 JTAG_TMS PTA4
3 4 JTAG_TCLK PTC4
5 6 JTAG_TDO PTA10
7 8 JTAG_TDI PTC5
9 10 JTAG_TRST PTA5
C15 0.1 µF
HDR 2X5
aaa-035485

Figure 5. MCU

3.7.4 MC33664 schematic

See the MC33664 data sheet for the schematic. https://www.nxp.com/docs/en/data-sheet/MC33664_SDS.pdf

3.8 Application reference

3.8.1 TPL based architecture

For an electric car or bus application, the trend is to use a mixed centralized-distributed architecture. The RD33771CNTREVM could be set up to meet these application needs. Here is a TPL daisy-chain-based BMS communication example:

- One centralized board in battery pack
- Capacitor or transformer isolation on-board
- Transformer isolation off-board
- Daisy chain linked each battery pack
- Loopback is optional, for communication robust
- The MCU module of each board need configured bypass, no use for this case.

This board provides a reference design for four BCC and one MCU.

- Isolated communication with a transformer (off-board)
- Capacitor isolation (on-board) support up to 40 m between each node
- Loopback chain can be used to increase the communication robustness

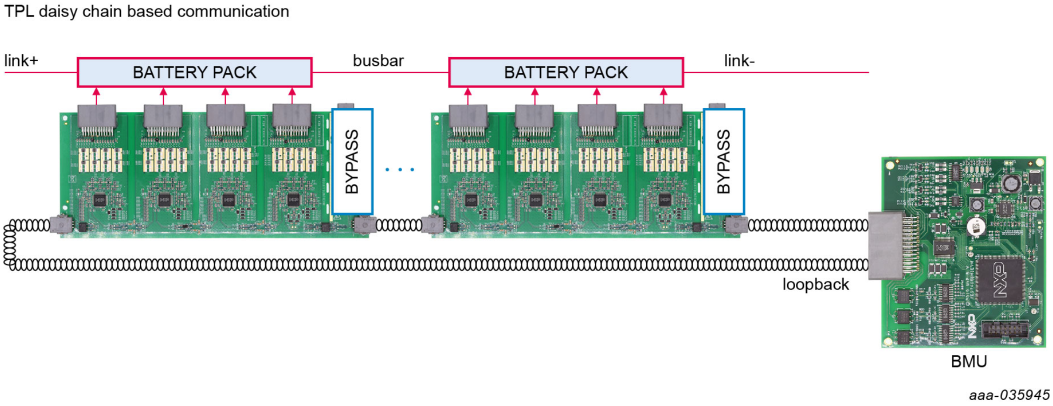


Figure 6. TPL Daisy chain based communication

3.8.2 CAN based architecture

In earlier BMS architecture customer still need CAN communication between each battery pack controlled by BMU, or use a standalone centralized board for eCar application, here is CAN based BMS communication example: if it used it as standalone, the BMU could be removed and use onboard MCU transfer AFE data directly to VCU.

- One centralized board in battery pack
- Capacitor or transformer isolation on-board
- Each MCU of this board manage 4 BCC on the board
- Each MCU collect BCCs data and transfer to BMU by CAN

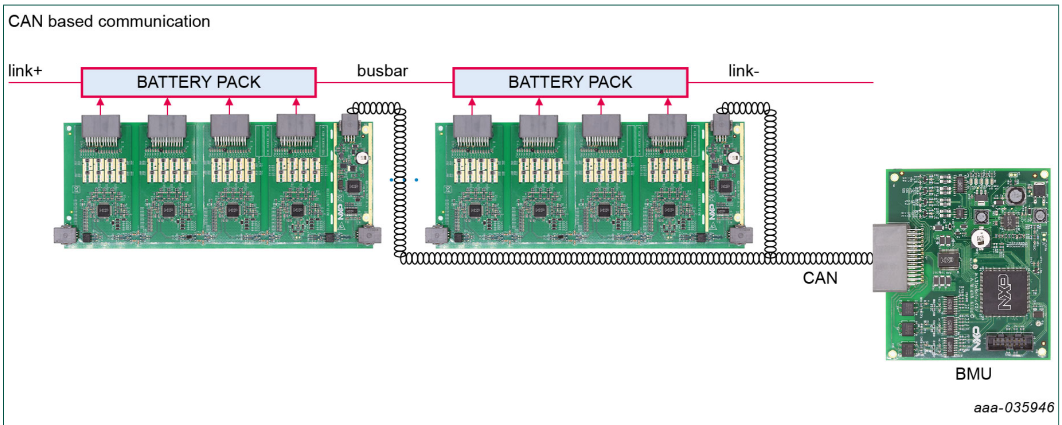


Figure 7. CAN based communication

4 Configuring the hardware

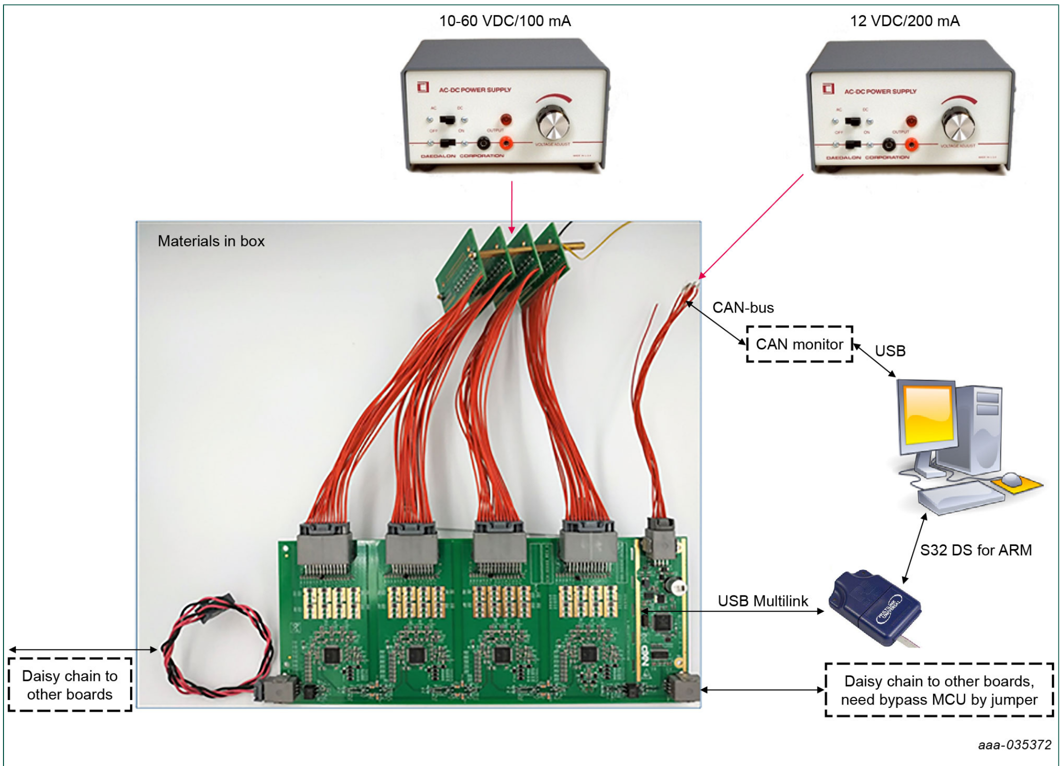


Figure 8. Typical system configuration

Setup steps:

1. Plug battery simulator and low-voltage cables to the board as shown
2. Connect battery simulator cable to a power supply #1 (10 to 60 VDC, current limitation 100 mA). Pin mask at bottom of board.
   Caution: Incorrect connections of the power and ground wires will damage the board.
3. Connect the low-voltage cable to power supply #2 (12 VDC, current limitation 200 mA), Pin mask at bottom of board.
   Caution: Incorrect connections of the power and ground wires will damage the board.
   Notes:

- If successfully supplied, the LED on the board will light. The power consumption for:power supply #1 is about 40 mA, power supply #2 about 50 mA.
- If no MCU program is downloaded or the MCU power is off, the LED on each BCC will turn off after 60 seconds.

4. Connect multilink to the JTAG port next to the MCU.
5. Launch S32DS for ARM in PC
6. Import example software project, launch download/debug in S32DS for ARM to evaluate this board.

5 Installing software and tools

5.1 Installing S32 Design Studio IDE for ARM

The S32 Design Studio IDE is a complimentary integrated development environment for automotive and ultra-reliable MCUs that enables editing, compiling and debugging of designs.

1. Install S32 Design Studio for ARM (Version 2018.R1 is recommended).
   Note: Registration is required.
2. Download the desired software package from the RD33771CNTREVM product information page [1].

5.2 Get the example project

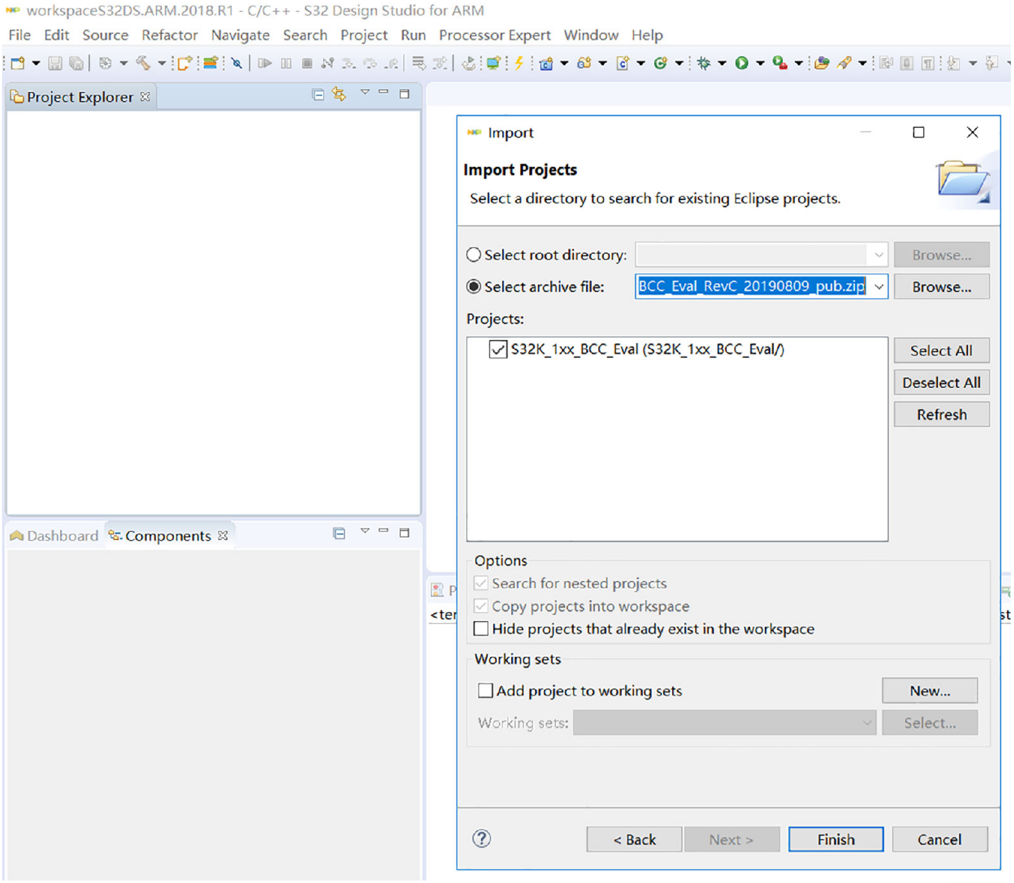
Open the RD33771CNTREVM page, go to the design column to get the source code package

(here should provide a page link)

5.3 Start development with an example source code

In S32 Design Studio for ARM, import the software project.

1. From the **File** menu, select **Import**.
2. Choose **General** > **Existing Project into Workspace**, and then click **Next**.
3. Click **Select archive file**. Click **Browse**, then locate project from step 2.
4. Select **S32K_1xx_BCC_Eval_RevC_20190809_pub.zip** project, and then click **Finish**.


workspaceS32DS.ARM.2018.R1 - C/C++ - S32 Design Studio for ARM
File Edit Source Refactor Navigate Search Project Run Processor Expert Window Help
Project Explorer
Dashboard
Components
Import
Browse...
Browse...
Select All
Deselect All
Refresh
New...
Select...
< Back
Next >
Finish
Cancel

Import Projects

Select a directory to search for existing Eclipse projects.

○ Select root directory:

● Select archive file: BCC_Eval_RevC_20190809_pub.zip

Projects:

☑ S32K_1xx_BCC_Eval (S32K_1xx_BCC_Eval/)

Options

☑ Search for nested projects

☑ Copy projects into workspace

□ Hide projects that already exist in the workspace

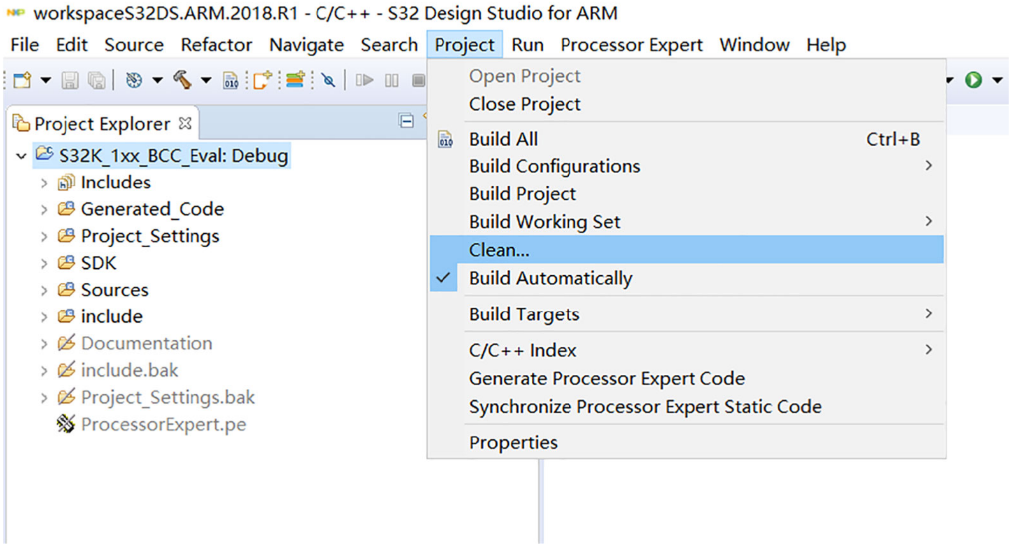
Working sets

□ Add project to working sets

Working sets:

aaa-035486

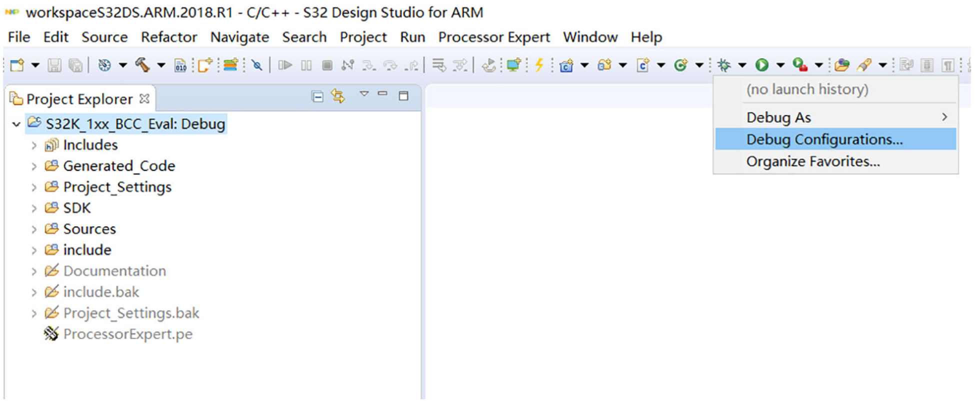
5. Select S32K_1xx_BCC_Eval_RevC_20190809_pub.zip project, and then click **Project** > **Clean ...** / **Build Project** to rebuild the project.



aaa-035487

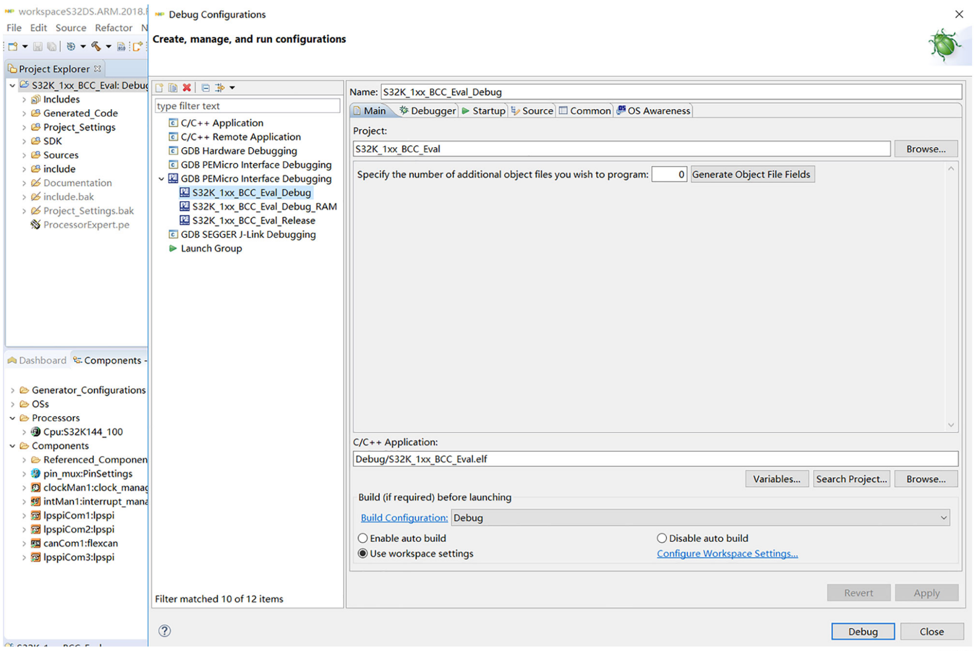
6. Debug the software project.

a. Go to **Run** > **Debug Configurations**.



aaa-035488

b. Choose **GDB PEMicro Interface Debugging** > **S32K_1xx_BCC_Eval_Debug**.



Name: S32K_1xx_BCC_Eval_Debug

Project: S32K_1xx_BCC_Eval

Specify the number of additional object files you wish to program: 0

C/C++ Application: Debug/S32K_1xx_BCC_Eval.elf

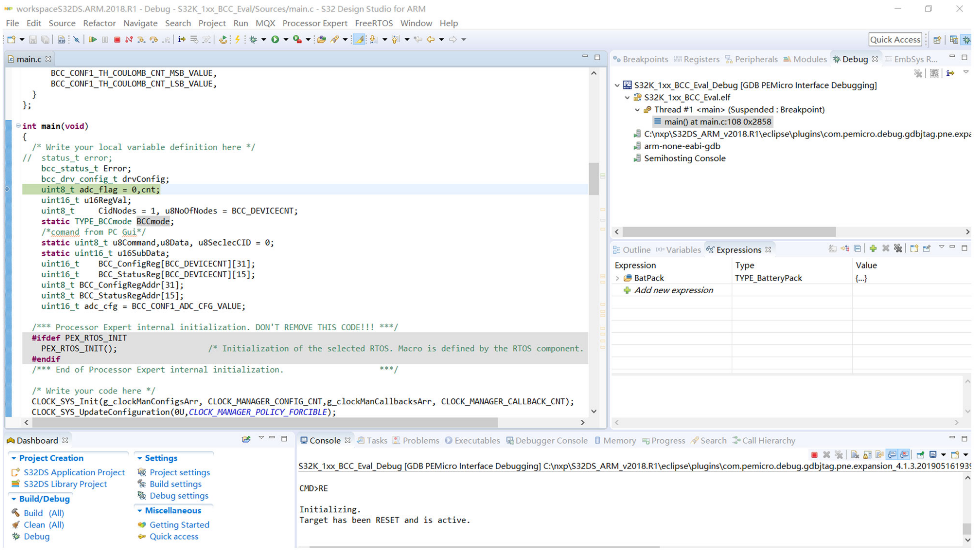
Build (if required) before launching

Build Configuration: Debug

○ Enable auto build
○ Disable auto build
● Use workspace settings
Configure Workspace Settings...

aaa-035489

c. Click the **Debug** button to download the firmware and start debugging



workspaceS32DS.ARM.2018.R1 - Debug - S32K_1xx_BCC_Eval/Sources/main.c - S32 Design Studio for ARM
File Edit Source Refactor Navigate Search Project Run MQX Processor Expert FreeRTOS Window Help
Quick Access
main.c
Breakpoints Registers Peripherals Modules Debug EmbSys R...
Outline Variables Expressions
Dashboard
Project Creation
S32DS Application Project
S32DS Library Project
Build/Debug
Build (All)
Clean (All)
Debug
Settings
Project settings
Build settings
Debug settings
Miscellaneous
Getting Started
Quick access
Console Tasks Problems Executables Debugger Console Memory Progress Search Call Hierarchy

```
        BCC_CONF1_TH_COULOMB_CNT_MSB_VALUE,
        BCC_CONF1_TH_COULOMB_CNT_LSB_VALUE,
    }
};

int main(void)
{
  /* Write your local variable definition here */
//  status_t error;
    bcc_status_t Error;
    bcc_drv_config_t drvConfig;
    uint8_t adc_flag = 0,cnt;
    uint16_t u16RegVal;
    uint8_t     CidNodes = 1, u8NoOfNodes = BCC_DEVICECNT;
    static TYPE_BCCmode BCCmode;
    /*comand from PC Gui*/
    static uint8_t u8Command,u8Data, u8SeclecCID = 0;
    static uint16_t u16SubData;
    uint16_t    BCC_ConfigReg[BCC_DEVICECNT][31];
    uint16_t    BCC_StatusReg[BCC_DEVICECNT][15];
    uint8_t BCC_ConfigRegAddr[31];
    uint8_t BCC_StatusRegAddr[15];
    uint16_t adc_cfg = BCC_CONF1_ADC_CFG_VALUE;

  /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
  #ifdef PEX_RTOS_INIT
    PEX_RTOS_INIT();                 /* Initialization of the selected RTOS. Macro is defined by the RTOS component.
  #endif
  /*** End of Processor Expert internal initialization.                    ***/

  /* Write your code here */
  CLOCK_SYS_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT,g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
  CLOCK_SYS_UpdateConfiguration(0U,CLOCK_MANAGER_POLICY_FORCIBLE);
```

```
S32K_1xx_BCC_Eval_Debug [GDB PEMicro Interface Debugging]
  S32K_1xx_BCC_Eval.elf
    Thread #1 <main> (Suspended : Breakpoint)
      main() at main.c:108 0x2858
  C:\nxp\S32DS_ARM_v2018.R1\eclipse\plugins\com.pemicro.debug.gdbjtag.pne.expa
  arm-none-eabi-gdb
  Semihosting Console
```

Expression	Type	Value
BatPack	TYPE_BatteryPack	{...}
Add new expression		

```
S32K_1xx_BCC_Eval_Debug [GDB PEMicro Interface Debugging] C:\nxp\S32DS_ARM_v2018.R1\eclipse\plugins\com.pemicro.debug.gdbjtag.pne.expansion_4.1.3.20190516193

CMD>RE

Initializing.
Target has been RESET and is active.
```

aaa-035490

6 Get to know example code

6.1 Software flow

This example code provides a general BCC operation flow as following:

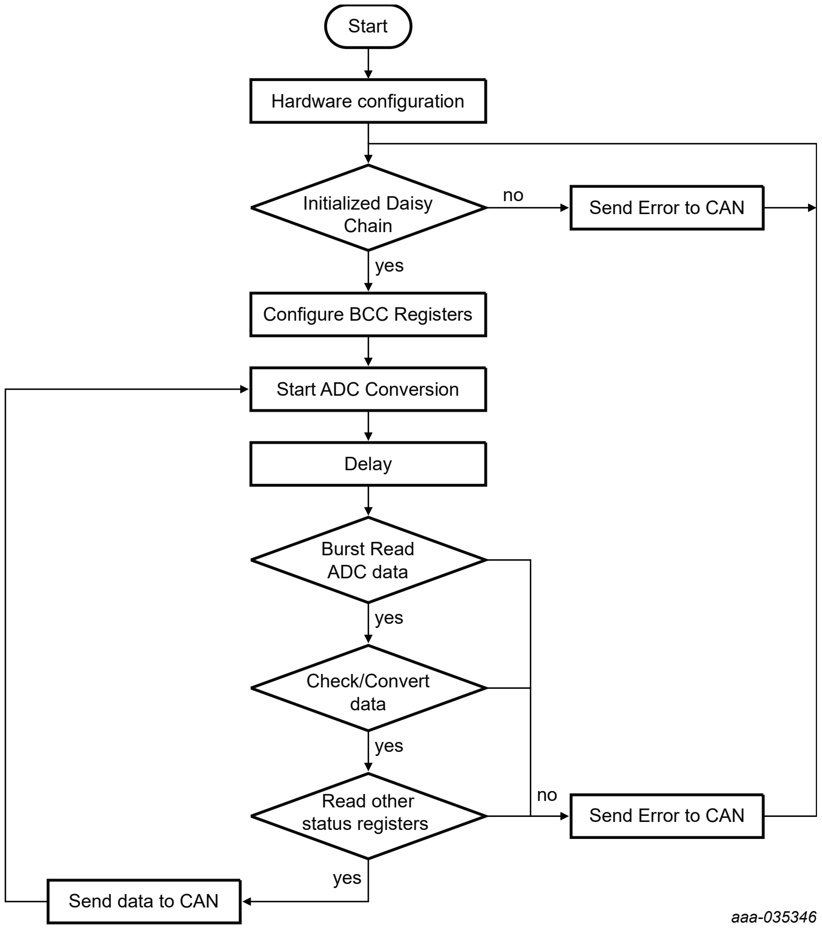


Figure 9. BCC operation flowchart

6.2 Variable definition

The most frequently used variables:

- The microcontroller defines how many nodes are in the daisy chain

```
#define BCC_DEVICECNT    4U
```

- BCC configuration structure – defined daisy chain parameters

```
typedef struct {
    uint8_t drvInstance;                     /*!< BCC driver instance. Passed to the external
                                                  functions defined by the user. */
    bcc_mode_t commMode;                     /*!< BCC communication mode. */
    uint8_t devicesCnt;                      /*!< Number of BCC devices. SPI mode allows one
                                                  device only, TPL mode allows up to 15 devices. */
    bcc_device_t device[BCC_DEVICE_CNT_MAX]; /*!< BCC device type of
                                                  [0] BCC with CID 1, [1] BCC with CID 2, etc. */
    uint16_t cellCnt[BCC_DEVICE_CNT_MAX];    /*!< Number of connected cells to each BCC.
                                                 [0] BCC with CID 1, [1] BCC with CID 2, etc. */

    bcc_comp_config_t compConfig;            /*!< Configuration of external components. */
    bcc_drv_data_t drvData;                  /*!< Internal driver data. */
            } bcc_drv_config_t;
```

- BCC data structure – defined ADC/status registers and converted values

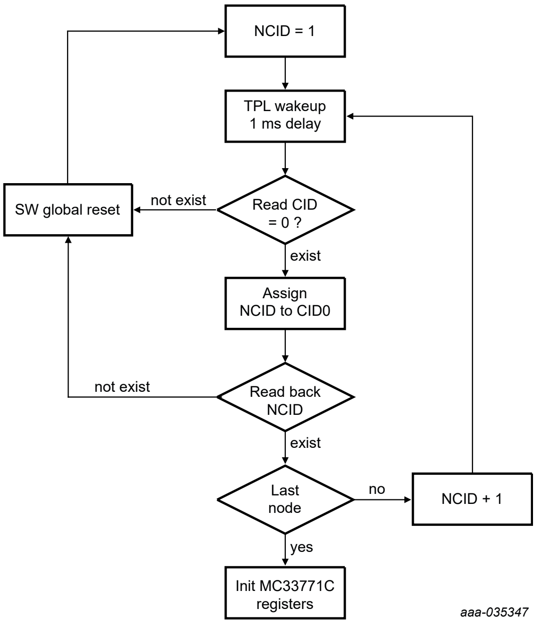
```
typedefstruct {
    uint16_t u16VoltCell[15][22];    /*!<1 stack voltage, 14 CT voltage and 7 AN voltage
                                           measured values in [mV]. */
    uint16_t u16RawCell[15][25];     /*!<1 stack voltage, 14 CT voltage , 7 AN voltage, 1 IC
                                           temperature ADC1-A and ADC1-B voltage raw values
                                           from register*/
    uint16_t u16Temp[15];            /*!<1 IC temperature measured value in [°C]  */
    uint16_t u16VbgADC1[15][2];      /*!<ADC1-A and ADC1-B voltage measured value in [mV] */
    int32_t i32Current[15];          /*!signed current shunt voltage measured value in [mV]  */
    int32_t i32RawCurrent[15];       /*!unsigned current shunt voltage raw value from register*/
    uint16_t u16CCSamples[15];       /*!Number of samples in coulomb counter */
    uint32_t s32CCCounter[15];       /*!signed Coulomb counting accumulator*/
    uint16_t u16SiliconRev[15];      /*!BCC IC Silicon revision */
    uint16_t u16FautStatus[15][11];  /*!BCC IC fault and status register data */
             }TYPE_BatteryPack;
```

6.3 Functions description

6.3.1 Initialization

Function:	CID Assignment and registers Initialization
Key parameters	• The number of nodes • Initialization values of each MC33771C device register
Function name in example code:	BCC_Init
Description:	

- When an MC33771C device is powered up or after any reset, the device will enter INIT mode and its cluster ID (in INIT register) equals 0. Register INIT at address 0x01 is the only writeable register and is used to configure the CID.
- In SPI mode, the initialization only needs to write INIT[CID] = 1.
- In TPL mode, assign a different CID to each MC33771C by writing CID into register INIT by local write command.
- After assigning CID, the MC33771C device will go to normal mode.
- A proposed initialization in TPL mode flow is shown in the figure below.



It is recommended to configure SYS_CFGx, ADC_CFG, GPIO_CFGx, FAULT_MASKx and WAKEUP_MASKx registers after CID assignment. If used, the threshold of OV/UV, OT/UT also needs to be configured at this phase.

6.3.2 Register operation

Function:	Read registers command	Write register command	Global write command	Update register command
Key parameters	• CID • Register address • Number of registers • Registers content	• CID • Register address • Register value	• Register address • Register value	• CID • Register address • Bits to be updated • New update value
Function name in example code:	BCC_ReadRegisters	BCC_WriteRegister	BCC_ WriteRegisterGlobal	BCC_UpdateRegister
Description:	This function reads values from addressed register from one specified device.	This function writes a value to addressed register of selected device.	This function writes a value to addressed register of all devices.	This function updates content of a selected register. It affects bits specified by a bit mask only.

6.3.3 Measurement

Function:	Start ADC conversion
Key parameters	• ADC configuration

Function name in example code:	BCC_StartConversion
Description:	
This function starts ADC conversion for all MC33771C in daisy-chain to convert the cell voltages, the stack voltage, the GPIOs used as analog inputs, the battery current (if set SYS_CFG1[I_MEAS_EN]), and the internal channel voltage. Parameter is used to configure the number of average and ADC resolution. In TPL mode, using a global command by writing ADC_CFG[SOC] = 1 will make all MC33771C devices start ADC conversion at the same time. In SPI mode, only a local command is used. Parameter ADC configuration is the value will be written in ADC_CFG. For example, in order to finish ADC conversion in 10ms, it is recommended to writing ADC_CFG[AVG] = 0b0100, ADC_CFG[ADC1_x_DEF] = 0b11.	

Function:	Check status of conversion
Key parameters	• CID • Returned Check Result
Function name in example code:	BCC_IsConverting
Description:	
This function checks status of conversion defined by [EOC_N] bit in ADC_CFG register. Users can use this function to check if commanded conversion is completed by polling ADC_CFG[EOC_N] bit until it equals 0. Users also can give up this function and just wait for the maximum conversion time which equals to 546 μs in 16-bit mode after sending a conversion command.	

Function:	Get raw values of measurement result
Key parameters	• CID • Returned measurement result
Function name in example code:	BCC_GetRawMeasurements
Description:	
This function reads the measurement registers and returns raw values. When a conversion is completed, the measurement results shall be stored in the registers that are linked to the measurements (addresses $2D to $4A). MEAS_xxx registers(addresses $30 to $4A) all contain DATA_RDY bits (bit 15), which is needed to be removed and then stored as raw measurement values before decoding	

Function:	True value calculation
Key parameter	Returned measurement true value
Function name in example code:	BCC_DecodeRawMeasurements
Description:	
This function calculates the true measurement values from raw measurement results. Users can get true values by getting raw values and multiply by the resolution. VCT_ANx_RES is the resolution for cell terminals CTx, analog inputs ANx, and band gap diagnostic reference voltage, VVPWR_RES is the resolution for stack voltage, V2RES is the resolution for shunt resistor voltage, and IC_TEMP1_RES is the resolution for junction temperature. All of the detailed resolutions can be found in Electrical Characteristics section of the data sheet.	

6.3.4 Cell balancing

Function:	Activate/deactivate selected cell balancing driver
Key parameters	• CID • Selected cell balance channel number • Command of CBx driver on/off • Cell balance timer
Function name in example code:	BCC_SetCBIndividually
Description:	
This function is used to activate or deactivate a selected cell balancing driver by updating the CBx_CFG[CB_EN] bit. Before using this function to activate cell balance drivers, the SYS_CFG1[CB_DRVEN] bit must be set to logic 1. Users also can optionally configure individual cell balance timer (from 30 seconds to 511 minutes) through the CBx_CFG[CB_TIMER] bit when switching on CB driver.	

Function:	Checks status of each cell balance
Key parameters	• CID • Returned channel balance status
Function name in example code:	BCC_GetCBStatus
Description:	
This function checks status of each cell balance by reading the CB_DRV_STS register.	

Function:	Pause all channel cell balancing driver manually
Key parameters	• CID • Command pause/unpause
Function name in example code:	BCC_PauseCBDrivers
Description:	
This function can be used to disable the cell balance by setting SYS_CFG1[CB_MANUAL_PAUSE] to 1. After this bit is set again to logic 0, the cell balance switches are restored according to the balance status before pausing. It is recommended to use this function to disable the cell balance before on-demand conversion for a more precise measurement result. Leakage and CB circuit open diagnostic must disable cell balance for an accurate result. **Note:** the cell balance timers are not frozen during a manual pause.	

6.3.5 Fault status operation

Function:	Read FAULT related registers
Key parameters	• CID • Returned fault status Information
Function name in example code:	BCC_GetFaultStatus
Description:	

This function is used to read some registers of fault status. Users may use this function to get the following faults: • CT overvoltage fault (from register CELL_OV_FLT) • CT undervoltage fault (from register CELL_UV_FLT) • CB pin or resistor open fault (from register CB_OPEN_FLT) • CB circuit short fault (from register CB_SHORT_FLT) • AN undertemperature and overtemperature fault (from register AN_OT_UT_FLT) • GPIO pin short and analog inputs open load detection (from register GPIO_SHORT_ANx_OPEN_STS) • Number of communication errors detected (from register COM_STATUS) • Fault types 1 (from register FAULT1_STATUS) • Fault types 2 (from register FAULT2_STATUS) • Fault types 3 (from register FAULT3_STATUS)

Function:	Clear FAULT status registers
Key parameter	• CID
Function name in example code:	BCC_ClearFaultStatus
Description:	
This function is used to clear fault status registers (refer to the previous table). Users may use this function to clear fault after initialization or before and after diagnostics to avoid false detections.	

7 Import and debug embedded software

After configuring the hardware, set up the software using the following procedure.

1. Install S32 Design Studio for ARM (Version 2018.R1 is recommended). See Section 5.
2. Download the desired software package from the RD33771CNTREVM product information page [1].

8 Set up evaluation GUI (optional)

The RD33771CNTREVM evaluation board provides a CAN bus interface and embedded software integration with CAN communication protocol. Using a Graphical User Interface (GUI) on a PC workstation can help users perform functions more clearly and more easily than using a debug tool (S32 Design Studio IDE).

Users can develop a GUI by following the communication protocol and with the help of the third party USB-CAN tool and GUI development tool. The CAN0 port in J12 is used as an evaluation communication interface. Users must connect CAN0_high and CAN0_low to USB-CAN tool correctly. The CAN communication protocol is shown in Table 11.

The CAN setup is:

- CAN format : Extended
- baud rate : 500kbps

Notes:

- *Host is the device that sends the CAN message.*

- *All voltage, current and temperature data are raw values from MC33771 registers. Users need to convert them to actual values by multiplying by the resolution, which can be found in the MC33771 product data sheet.*

Table 11. CAN communication protocol

Host	Extended ID	Data							
		Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 4	Byte 5	Byte 6
Evaluation board	0x18801100	Stack voltage		Cell14 voltage		Cell13 voltage		Cell12 voltage	
Evaluation board	0x18801104	Cell11 voltage		Cell10 voltage		Cell9 voltage		Cell8 voltage	
Evaluation board	0x18801108	Cell7 voltage		Cell6 voltage		Cell5 voltage		Cell4 voltage	
Evaluation board	0x1880110C	Cell3 voltage		Cell2 voltage		Cell1 voltage		AN6 voltage	
Evaluation board	0x18801110	AN5 voltage		AN4 voltage		AN3 voltage		AN2 voltage	
Evaluation board	0x18801114	AN1 voltage		AN0 voltage		IC temperature		n.a.	
Evaluation board	0x18802100	Current measurement				n.a.		n.a.	
Evaluation board	0x18804100	MC33771 FAULT1_STATUS		MC33771 FAULT2_STATUS		MC33771 FAULT3_STATUS		MC33771 COM_STATUS	
Evaluation board	0x18803100	Communication result		n.a.		n.a.		n.a.	
PC GUI	0x18800000	Command	Parameter	n.a.		n.a.		n.a.	
		0xC1 Reset MC33771	n.a.						
		0xC2 Initialize MC33771	1						
		0xC4 MC33771 go to sleep	n.a.						
		0xC5 MC33771 wake up	n.a.						
		0xC7 Crash signal control	0x01:generate 0x00:cancel						
		0xC8 High-side switch control	0x11: relay and fan on 0x10:fan on 0x01:relay on 0x00:off						

9 EMC performance

This is a summary of an EMC test report based on the RD33771CNTREVM board. Following chapter will list details. Customer's real application is variable. this is a general usage that could be referred by customer. This EMC test could also be used to evaluate the hardware/software design could archive.

Application	EMS (Criteria: No comm err, Cell voltage err ≤ 6 mV, GPIO err ≤ 16 mV)			EMI (Criteria : CISPR25)	
	BCI ISO 11452-4	ESD ISO 10605	RI ISO 11452-2	RE CISPR 25	CE CISPR 25
Mixed Architecture (3 boards, 4-AFE)	200 mA PASS	Up to 15 KV	100 V/m PASS	Class5 PASS	Class3 PASS

Application	EMS (Criteria: No comm err, Cell voltage err ≤ 6 mV, GPIO err ≤ 16 mV)			EMI (Criteria : CISPR25)	
	BCI ISO 11452-4	ESD ISO 10605	RI ISO 11452-2	RE CISPR 25	CE CISPR 25
Capacitor isolation architecture (3 boards, 4-AFE)	200 mA PASS	Up to 15 KV	100 V/m PASS	Class5 PASS	Class3 PASS

10 References

Following are URLs where you can obtain information on related NXP products and application solutions:

[1] **Product summary page for RD33771CNTREVM: HV Battery management system reference design** — http://www.nxp.com/products/RD33771CNTREVM

[2] **Product summary page for MC33664: Isolated Network High-Speed Transceiver** — http://www.nxp.com/products/MC33664

[3] **Product summary page for MC33771: 14-Channel Li-ion Battery Cell Controller IC** — http://www.nxp.com/products/MC33771C

[4] **Product summary page for UJA1169: Mini high-speed CAN companion system basis chip** — https://www.nxp.com/products/power-management/system-basis-chips/mini-system-basis-chips-sbcs/mini-high-speed-can-companion-system-basis-chip:UJA1169LTK

[5] **Product summary page for S32K144: 32-bit Automotive General Purpose Microcontroller** — https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/s32k-32-bit-automotive-general-purpose-microcontrollers:S32K

[6] **Support page for S32DS-PA: S32DS-ARM: S32 Design Studio for Arm** — https://www.nxp.com/design/software/development-software/s32-design-studio-ide/s32-design-studio-for-arm:S32DS-ARM

[7] **NXP DocStore** — docstore.nxp.com

11 Revision history

Revision history

Revision number	Date	Description
1	20200107	Initial release

12 Legal information

12.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

12.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Suitability for use in automotive applications — This NXP Semiconductors product has been qualified for use in automotive applications. Unless otherwise agreed in writing, the product is not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an “as is” and “with all faults” basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — While NXP Semiconductors has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP Semiconductors accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

12.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

SafeAssure — is a trademark of NXP B.V.

Tables

Figures

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 7 January 2020
Document identifier: UM11310