UG10071

S32K376 BMU and VCU Integration User Guide

Rev. 1.1 — 15 January 2024 **User guide**

Document Information

Information	Content
Keywords	BMS system, VCU system, bootloader, HD OTA, S32k376, MC33774A, MC33665A
Abstract	BMS and VCU are integrated in one ECU based on S32K376 MCU. BMS system monitors battery voltage, temperature, fault status, etc. VCU sample simulated pedal position, gear, sensors, etc. Software is developed based on RTD. The solution is intended to provide a mechanism for easy customer evaluation of the Cortex-M7@320MHz four cores MCU and to facilitate BMS and VCU hardware and software development. OTA and bootloader functions are also integrated into this project which makes program upgrades feasible. FS2633 is used for power management and monitoring.

1 Introduction

The document is a hardware and software user manual for the powertrain domain controller reference design integrated with the BMS and VCU in one ECU, based on the S32K376 MCU. The BMS system monitors battery voltage, temperature, fault status, and so on. VCU samples simulated pedal position, gear, sensors, and so on. The software is developed based on RTD. The reference design provides a mechanism for easy customer evaluation of the Cortex-M7@320MHz four-cores MCU and facilitates BMS and VCU hardware and software development. OTA and bootloader functions are also integrated into this reference design, making program upgrades feasible.

1.1 Acronyms

Table 1 provides a list of acronyms used throughout this document.

Table 1. List of acronyms

Acronym	Description
PDC	Powertrain Domain controller
EV	Electric Vehicle
BMS	Battery Management System
VCU	Vehicle Control Unit
SBC	System Basis Chip
TPL	Transform Physical Layer
BCC	Battery Cell controller
RTC	Real-time Clock
DNP	Do Not Populate
CAN	Controller Area Network
ECU	Electronic Control Unit
MSDI	Multiple Switch Detection Interfaces
HSD	High Side Switch
LSD	Low Side Switch
PHY	Physical
OBC	On Board Charge
JTAG	Joint Test Action Group
LIN	Local Interconnect Network
PWM	Pulse Width Modulation

2 Hardware user guide

2.1 Overview

Figure 1 shows the whole hardware BMU board and lists each device type. See Table 2 for the connector of the hardware.

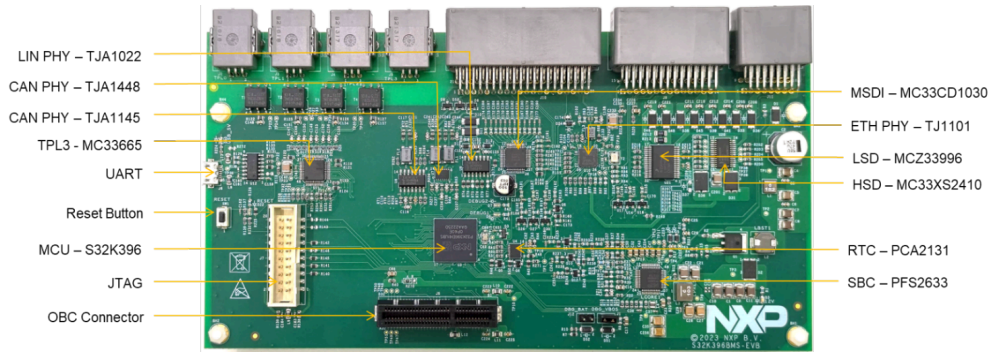


Figure 1. BMU board

Figure 2 shows the system block diagram.

- AFE MC33774 monitors battery status.
- Gateway MC33665 transfers the SPI signal to TPL3.
- SBC FS2633 supplies power for S32K376 and external devices.
- HSD, LSD, and MSDI are used for VCU functions.
- CAN, LIN, and UART communicate with other ECU.
- RTC is used to read real time for BMS Soc calculation.

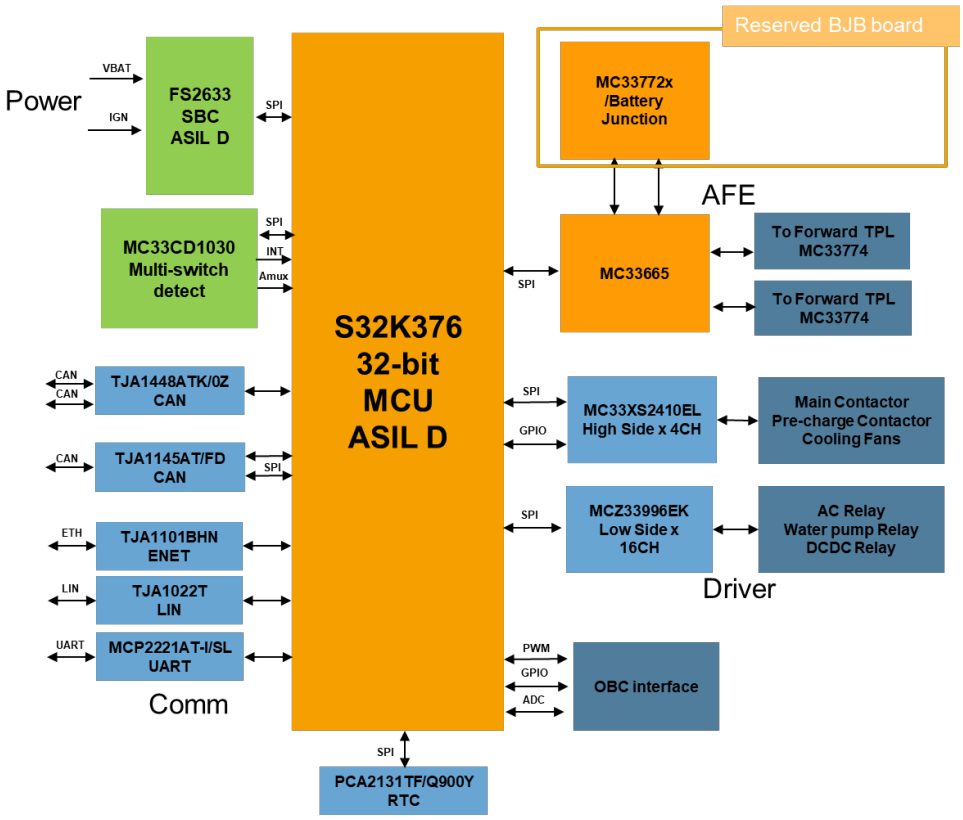


Figure 2. System block diagram

2.2 Board features

The system mainly consists of a BMU board and several CMU boards connected in series with a daisy chain with TPL3.

The following are the key features of the master board:

- NXP S32K376 microcontroller
- Integrate NXP Power FS2633 (ASIL D) as SBC
- 1 x MC33665 as TPL(support four TPL interfaces) to support daisy chain with loopback
- One user reset switch with reset status LEDs
- Two user LEDs are connected to the GPIO for test
- Standard 20-pin JTAG debug connector
- One automotive tiny real-time clock/calendar with alarm function
- Integrate multiple communication interfaces:
  - 1 x 100BASE-T1 Automotive Ethernet interface TJA1101
  - 3 x CAN interfaces
  - 1 x LIN interface
  - 1 x Mini USB/UART transceiver to interface with MCU
- Integrate input/output devices:
  - 10 x CH digital and 4 x CH analog multiple switch detection interfaces
  - 6 x CH analog signal directly input MCU for sampling pedal position, pressure, and so on
  - 6 x CH digital signal directly input MCU for sampling AC switch, gear, and so on
- ECU connector, routing external I/O signals including:
  - 6 x ADC input channels
  - 2 x PWM input capture channels
  - 5 x CH 5 V + 5 x CH 12 V power supply for external sensor
  - 4 x HSD output channels
  - 16 x LSD output channels
  - 4 x Daisy chain interface

2.3 Module introduction

2.3.1 Power SBC

2.3.1.1 Board power

The PFS2633 provides robust, scalable power management to the S32K376 MCU with Fail Silent safety monitoring measures that fit for ASIL D. It features multiple switch mode regulators and LDO voltage regulators to supply the microcontroller, sensors, peripheral ICs, and communication interface. It offers a high-precision voltage reference available to the system and a reference voltage for two independent voltage-tracking regulators. When external power, a typical 12 V automotive power supply, is applied to the J11 connector pin6-8 or pin14-16 of the BMU board, green power LED (D3) lights on. Figure 3 shows the whole BMU board power tree.

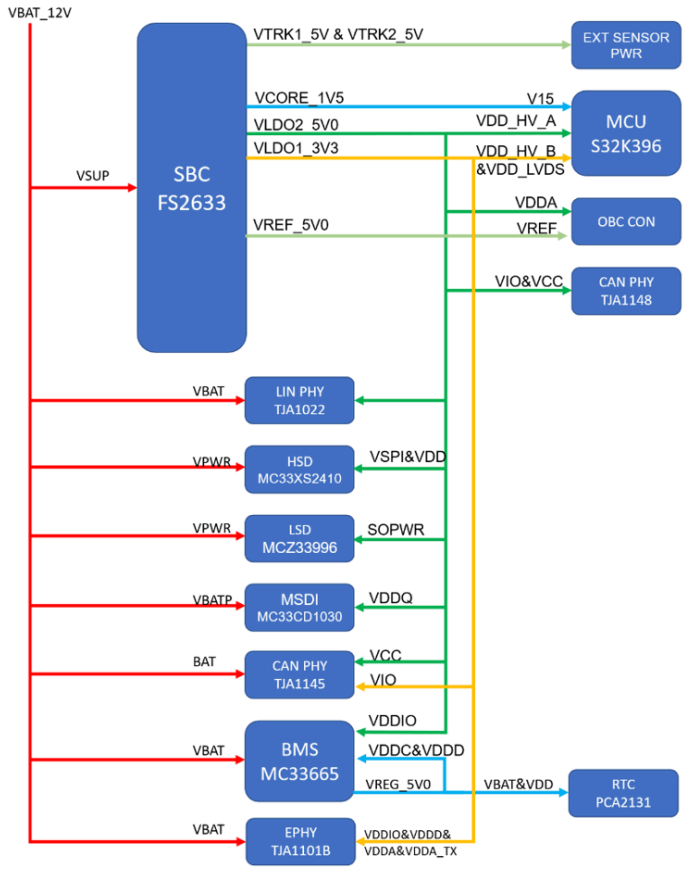


Figure 3. BMU power tree

2.3.1.2 Debug mode

The jumper J1 configures the Debug mode of the SBC, enable it to active the Debug mode, and then there is no deep fail-safe state. It is recommended to be closed to J1, as in Figure 4 at the development phase.

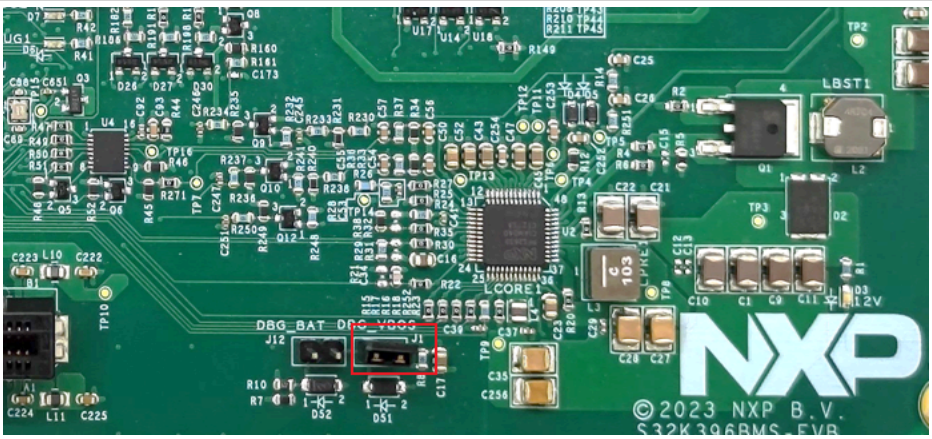


Figure 4. SBC DEBUG jumper

2.3.1.3 Reset circuit

When the specific condition is reached, the SBC resets MCU via the SBC_RSTB pin. As shown in Figure 5, the board supports the multiple MCU reset methods: SBC, JTAG, and Switch Reset(SW1).

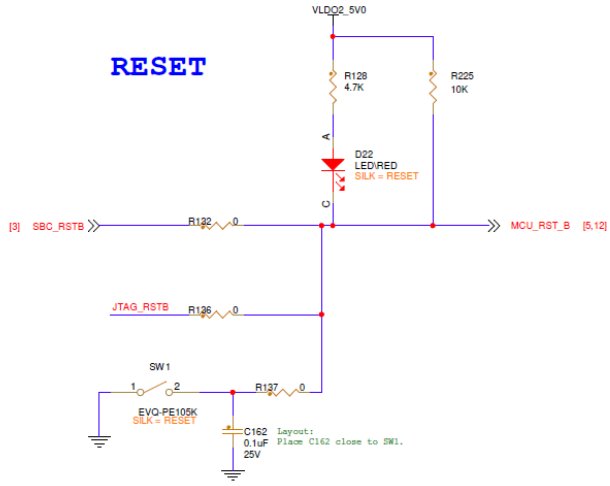


Figure 5. Board reset circuit

2.3.1.4 Communication interface

PFS2633(U2) has an LPSPI transceiver. It is routed to the MCU LPSPI_0 port, so only classic SPI is available via SBC SPI physical.

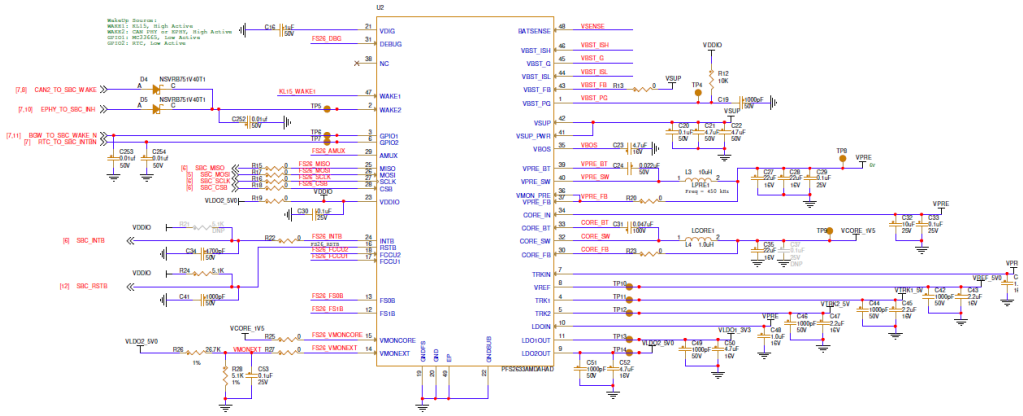


Figure 6. PFS2633 application circuit

2.3.2 MCU power and GND configuration

The VDD_HV_A is SDADC, SARADC, and SWG supply voltage connected to VLDO2 (5.0 V) from SBC, and VDD_HV_B is connected to VLDO1(3.3 V).

2.3.3 MCU external clock circuit

In addition to the internal fast 48 MHz and internal slow 32 kHz oscillator, the MCU can also be clocked by an external oscillator (Y1). The clock circuitry for the 40 MHz crystal is shown in Figure 7.

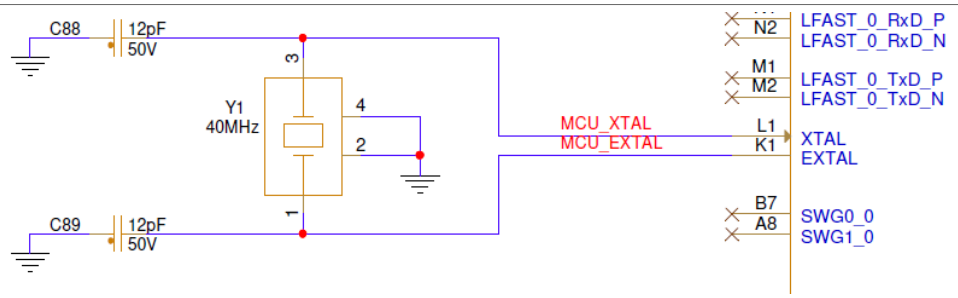


Figure 7. 40 MHz crystal circuit

2.3.4 JTAG connector

The BMU board is fitted with a 20-pin JTAG debug connector J7. Figure 8 shows the 20-pin JTAG connector pinout.

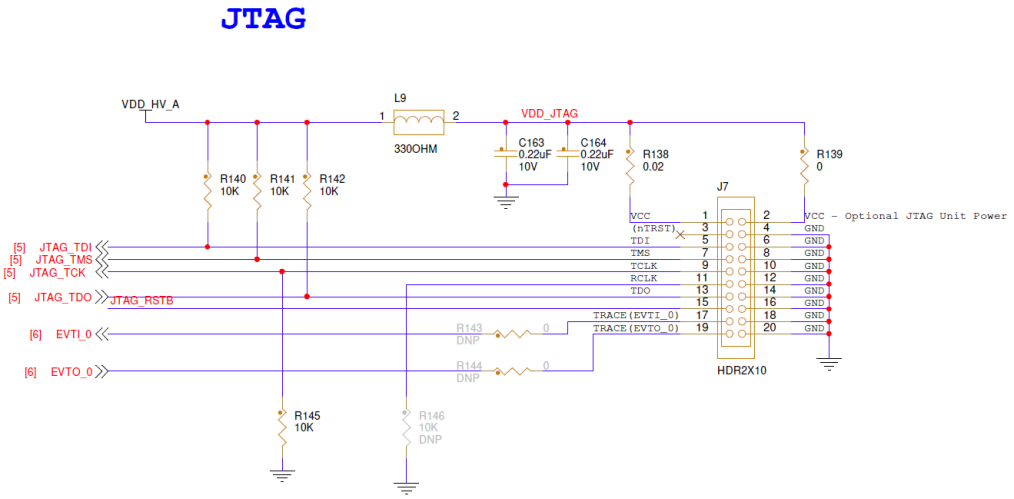


Figure 8. JTAG connector interface

2.3.5 USB/UART transceiver

Mini USB connector interfaced with MCP2221AT is connected to MCU port (M14, M12), USB to serial UART chip (U12). The USB to serial UART connection is shown in Figure 9 and can connect to a PC via mini USB to print or receive the debug strings by Serial Debug Assistant.

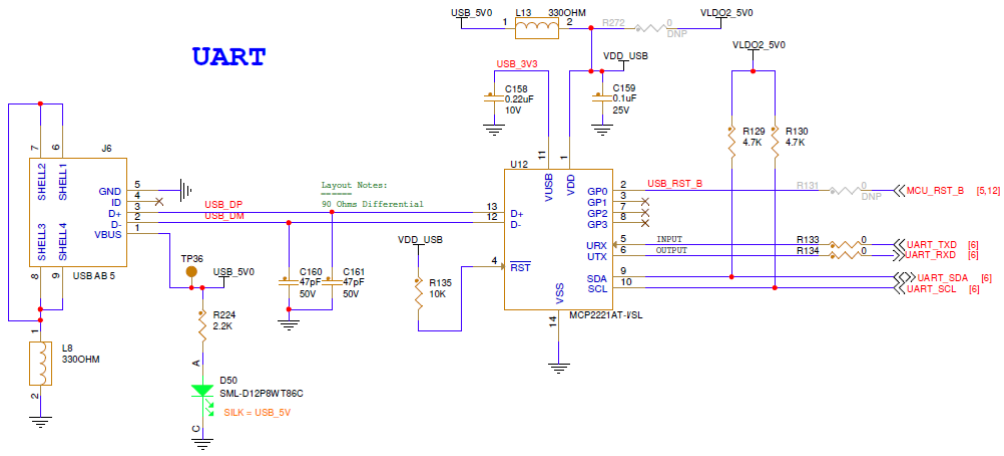


Figure 9. USB/UART transceiver circuit

2.3.6 CAN transceiver

The BMU board has two high speed CAN transceivers.

- TJA1448ATK/0Z (U7) is a galvanically isolated interface between the CAN controller and the physical two-wire CAN bus and is connected to the MCU CAN0/CAN1 port (J15, K14, F11, F10). The CAN0_STB and CAN1_STB pins control the enablement of the CAN instance, respectively, and levels must be high.
- TJA1145AT/FD/0Z(U8) is a high-speed CAN transceiver for partial networking, and connected to MCU CAN3 port (C4, C5). TJA1145AT/FD/0Z must drive through SPI. The SPI signal is transferred to the MCU through LPSPI3 with CS0. MCU uses the interface to configure the PHY.

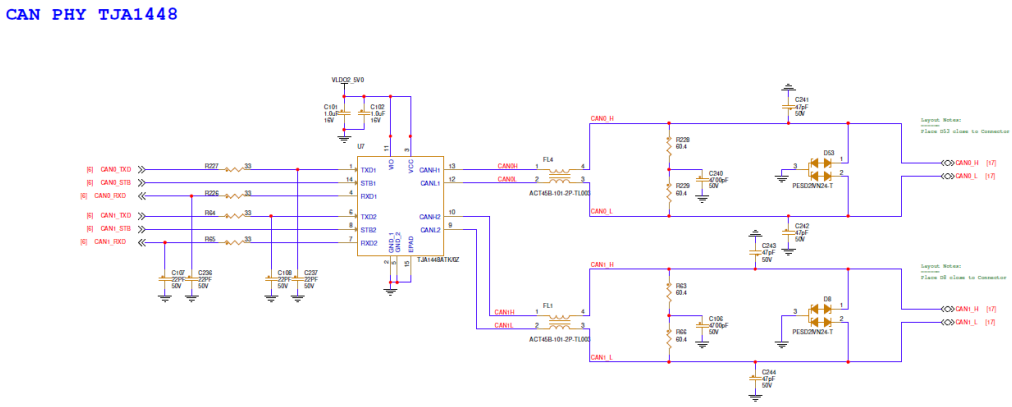


Figure 10. TJA1448ATK/0Z transceiver circuit

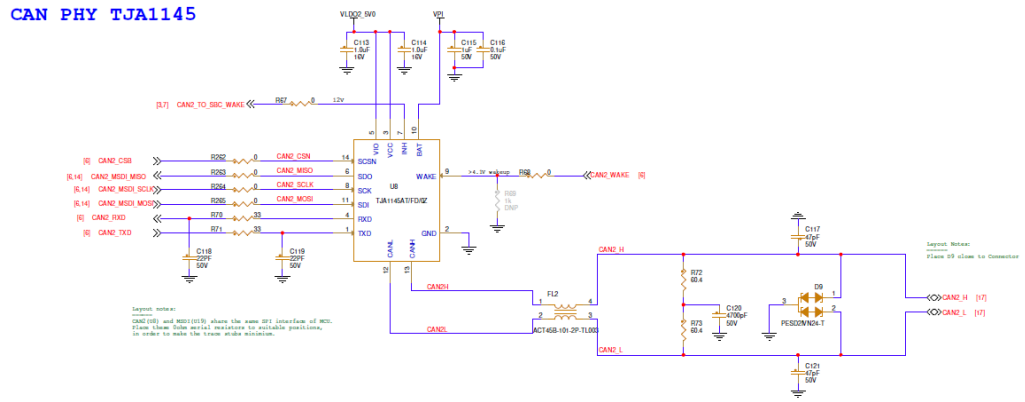


Figure 11. TJA1145AT/FD/0Z transceiver circuit

2.3.7 LIN transceiver

The BMU board is fitted with a TJA1022T LIN transceiver (U9) connected to MCU LPUART2 and LPUART3 ports (L13, M13, N13, N12). And the pins LIN_SLP1_N and LIN_SLP2_N must be high.

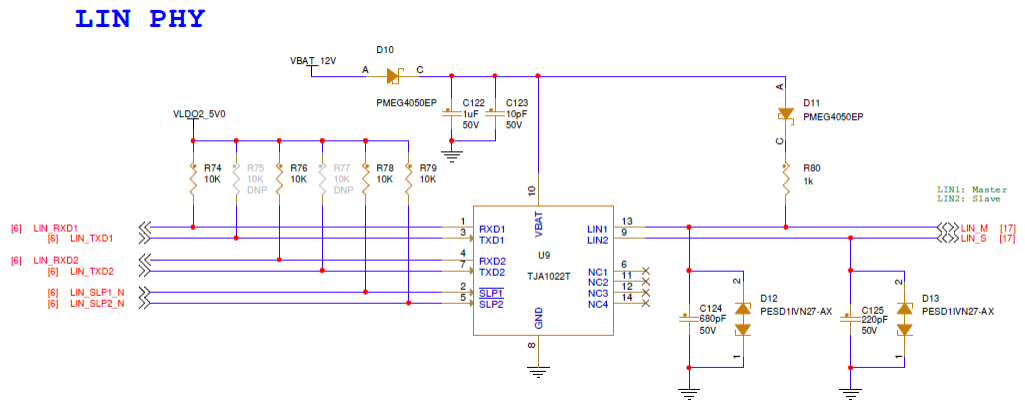


Figure 12. LIN transceiver circuit

2.3.8 Ethernet PHY

The master board includes support for the 100M-base T1 Automotive Ethernet PHY TJA1101 (U10), using the normal MII interface, is clocked by a 25 MHz external oscillator (Y2).

A differential signal pair (EPHY_TRX_P, EPHY_TRX_N) is routed to the J10 ECU connector.

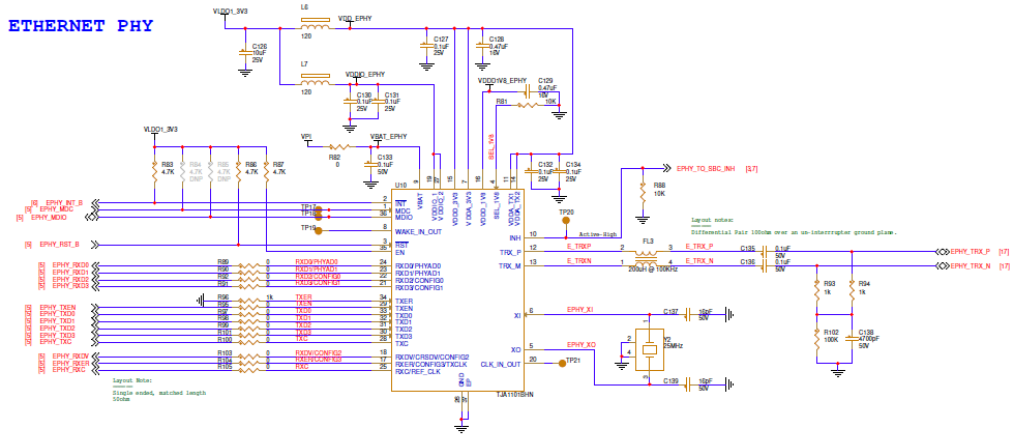


Table 11. Pin strapping configuration

Symbol	Pin	Value	Description
MASTER_SLAVE	22 (CONFIG0)	0	PHY configured as Slave
		1	PHY configured as Master
AUTO_OP	21 (CONFIG1)	0	managed operation
		1	autonomous operation
MII_MODE	17 (CONFIG3) 18 (CONFIG2)	00	Normal MII mode
		01	RMII mode (50 MHz input on REF_CLK)
		10	RMII mode (50 MHz output on REF_CLK)
		11	Reverse MII mode
PHYAD[2:1]	23 (PHYAD2)	-	bit 2 of PHY address used for the SMI
	24 (PHYAD1)	-	bit 1 of PHY address used for the SMI
LDO_MODE	4 (SEL_1V8)	0	internal 1.8 V LDO enabled
		1	external 1.8 V supply

Figure 13. TJA1101 ETH PHY circuit

2.3.9 SARADC interface

The BMU board enables 6x SARADC channels from MCU_AIN0 to MCU_AIN5. These channels are routed to connectors to measure external analog signals, such as pedals, and sensor signals, through ADC input signal process circuits (Figure 14).

In addition to MCU_AIN4 channel supports external 0-12 V analog signal input, other channels only support 0-5 V.

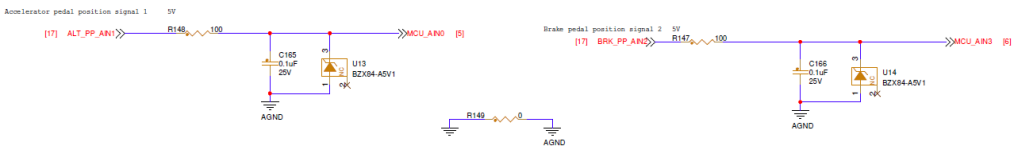


Figure 14. ADC input circuit

2.3.10 Digital signal interface

The BMU board supports detecting an external switching signal, such as gear signal, through a clamp circuit to the MCU port (MCU_DIN0-MCU_DIN5), as shown in Figure 15. In addition to MCU_DIN4 and MCU_DIN5 channels support 0/12 V digital signal input from J10, other channels only support 0/5 V.

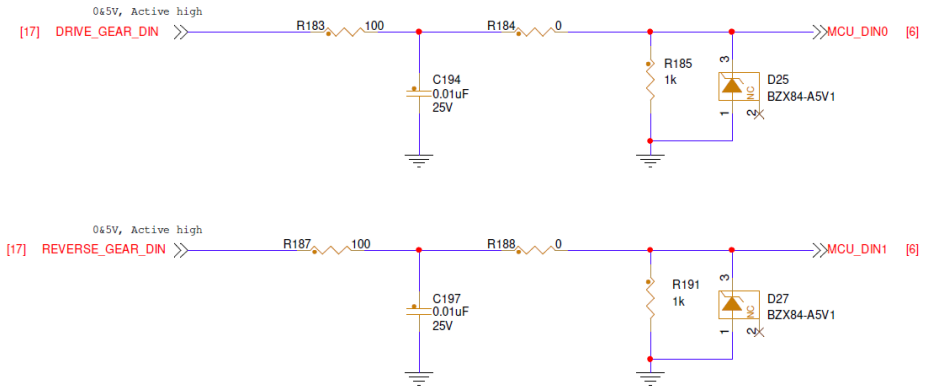


Figure 15. Digital input circuit

2.3.11 MSDI

The BMU board has an MC33CD1030 to detect the closing and opening of multiple switch contacts and analog input signals. The input signal is transferred to the MCU through LPSPI3 with CS1.

As shown in Figure 16, the BMU board enables SG0-SG9 as digital input, SP0-SP4 as analog input, this port connects to J10, and MCU read switch status via LPPSI3, analog signal via MSDI_AMUX through SARADC_0 channel P2.

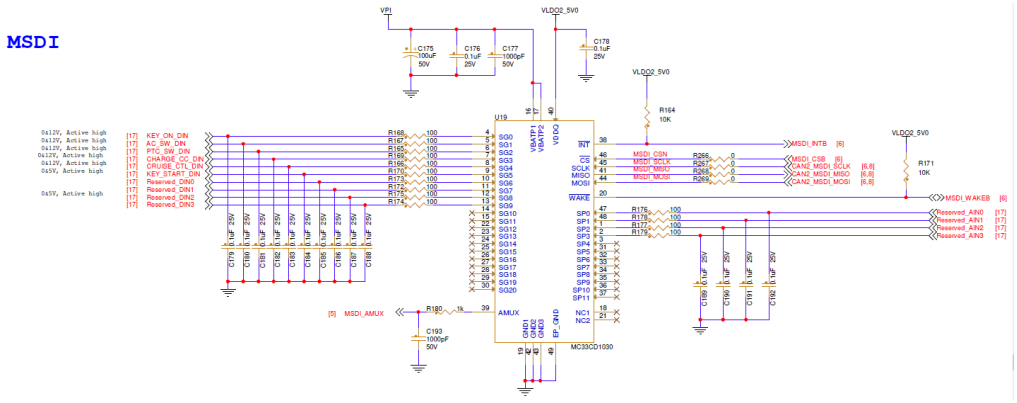


Figure 16. MSDI circuit

2.3.12 PWM input

The BMU board has two PWM input circuits to detect external PWM signal. The MCU_PWM_IN0/1 is routed to MCU eMIOS_0 channels 3/6. The user can use eMIOS mode - “Input Pulse Width Measurement(IPWM) mode” to measure the PWM signal.

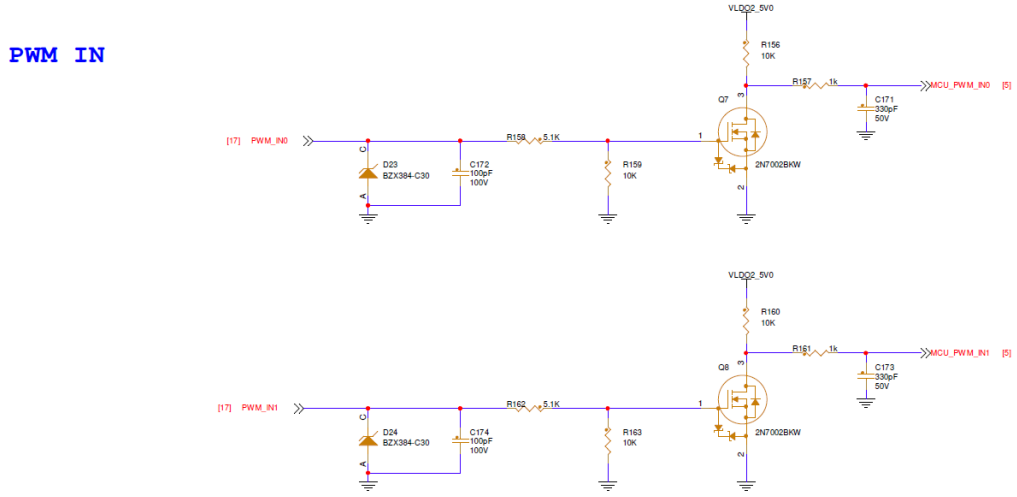


Figure 17. PWM circuit

2.3.13 HSD

The system requires several channels to drive an external load. Provide high current driving capability that could control these devices directly.

Generally load as:

- Main contactor - 12 V dc coil voltage rating, max 5 A coil current at ON.
- Precharge contactor - 12 V dc coil voltage rating, max 5 A coil current at ON.
- Cooling fans - 12 V dc, max 4.8 A coil current at ON.
- Heater - 12 V dc, max 5 A coil current at ON.

So, the BMU board uses an extreme switch device, NXP Quad high-side switch MC33XS2410EL(U20), to drive these loads. It is supplied by 12 V dc power, which could provide four channel high side outputs, each channel up to 5 A current.

MCU can configure LPSPI5 with CS1 to control it and also supports direct control of the output pins (OUT1-OUT4) via the HSD_IN1-HSD_IN4 with GPIO.

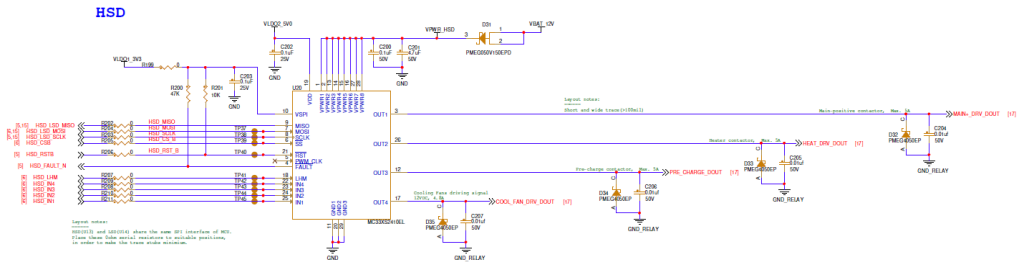


Figure 18. HSD circuit

2.3.14 LSD

The system requires several channels to control external contactors or relays, such as AC power switches, DCDC relays . The BMU board is fitted with an NXP 16-output low-side switch with a 24-bit serial input control MCZ33996 (U21), each outputting up to 1A current. MCU can configure LPSPI5 with CS2 to control it. The PWM pin – LSD_PWM is routed MCU eMIOS_0 channel 5, and the output of the PWM frequency is up to 2.0 kHz.

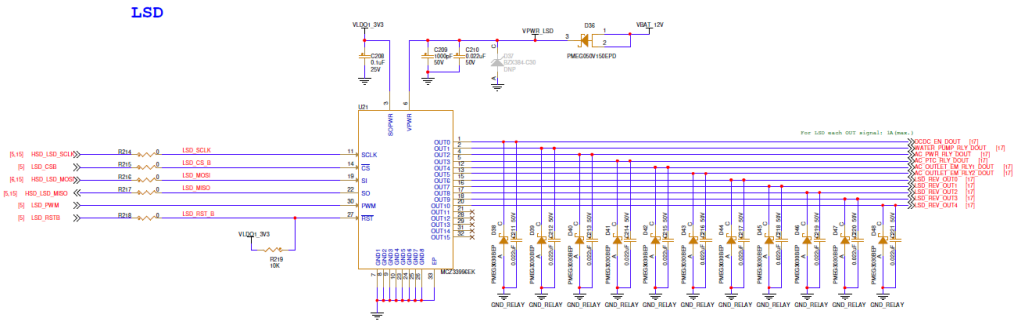


Figure 19. LSD circuit

2.3.15 TPL transceiver

Since the high voltage of battery packs, isolation is more important. This reference design isolates power supply and communications. The MC33665 is a transceiver physical layer transformer driver designed to interface an MCU conveniently to a high-speed isolated communication network. It supports both dual SPI and single SPI communication with MCU. The default dual SPI mode is configured on the BMU board. The customer can also switch to single SPI mode by mounting the DNP resistor.

The TPL communication port between the BMU board and CMU boards operates at a baud rate of 2 Mbit/s. It is linked as a daisy chain with a loopback circle in case one or more CMU boards break down. The MC33665 has four independent TPL daisy chain ports, and the protocol supports up to six TPL daisy chains, with 62 nodes per chain.

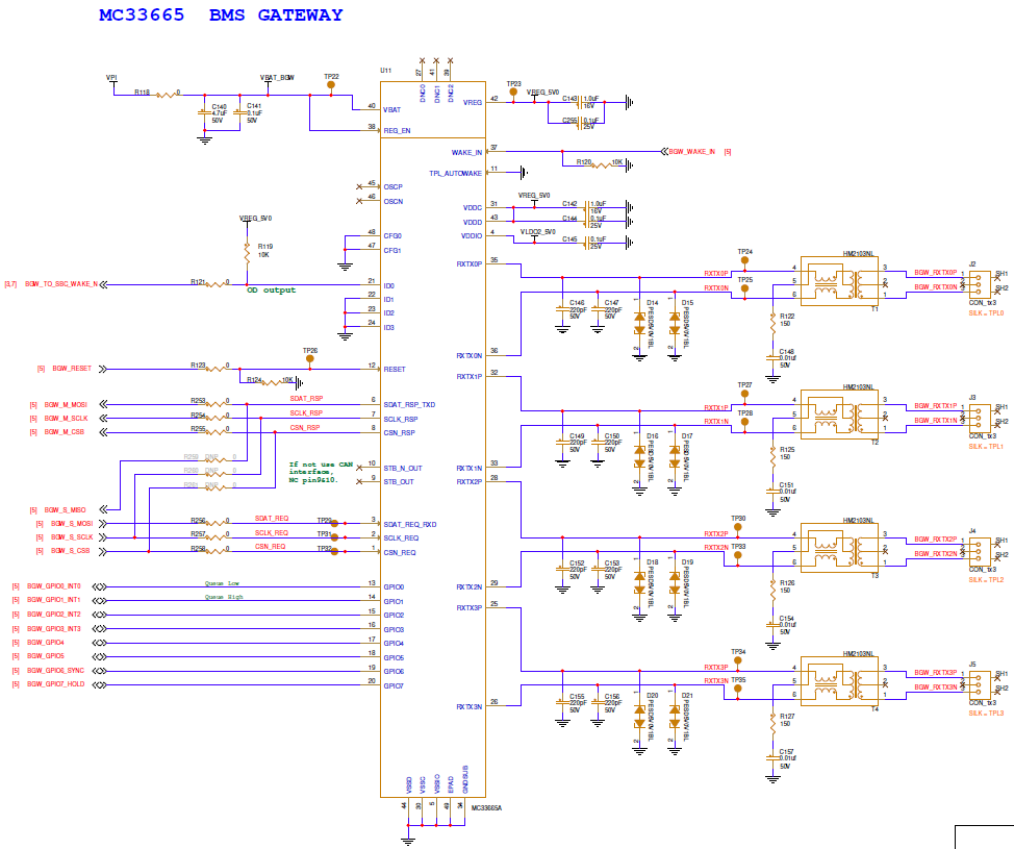


Figure 20. TPL circuit

2.3.16 RTC

The real-time clock is necessary for the system's two main functions:

- Provide an uninterrupted timer to the system as a reference. When the system is Low-Power mode, the time is recorded and statistical discharge.
- Provide the MCU cycle wake-up to check monitor data during normal or Low-Power mode.

The BMU board is fitted with an NXP automotive RTC PCA2131TF/Q900Y(U4) with SPI communication.

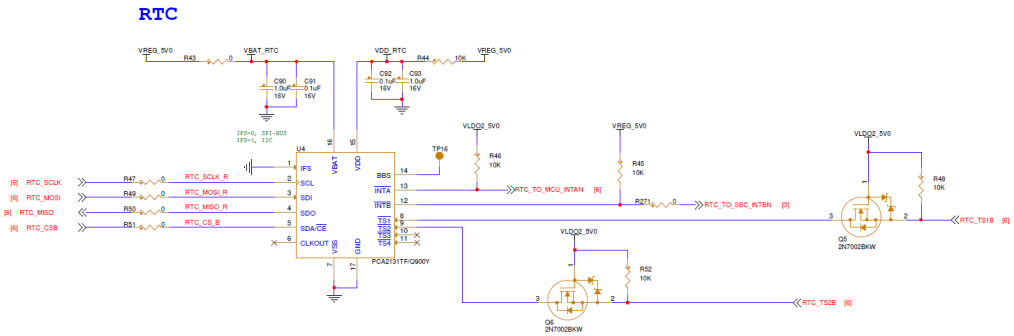


Figure 21. RTC circuit

2.3.17 OBC interface

In this reference design, the OBC interface is reserved. The control board and power board are separated. They are connected by a PCIe connector, which is shown in Figure 22. The PCIe interface includes 11x channels of the analog signal and 16x channels of the digital signal.

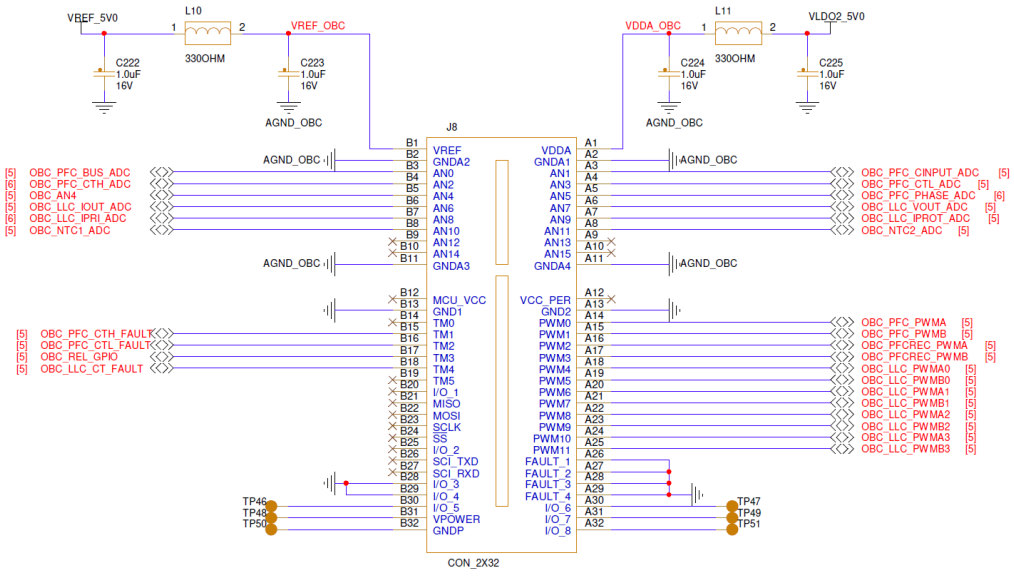


Figure 22. OBC interface

2.4 Hardware wake-up source

BMS usually requires Low-Power mode, this hardware design also reserved a multi wake-up method. Figure 23 shows the BMU board wake-up source.

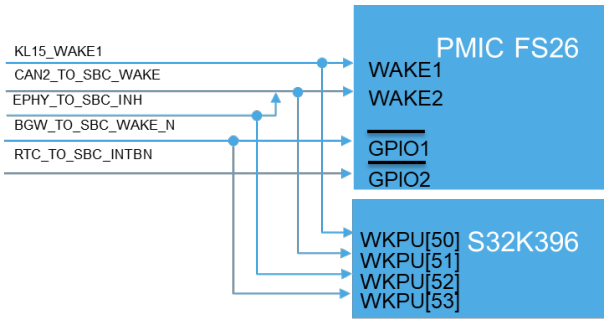


Figure 23. BMU board wake-up sources

2.5 ECU connector

Table 2 lists all the connections to the BMU board interface connector.

Table 2. Board interface connector details

Connector	Signal	Description	Connector	Signal	Description
J9-1	ALT_PS_5V0	5 V supply for Accelerator pedal	J9-15	CRUSH_SS_5V	5 V supply for crush sensor
J9-2	ALT_PS_5V0	5 V supply for Accelerator pedal	J9-16	GND	Digital GND
J9-3	GND	Digital GND	J9-17	BRK_PP_AIN1	Brake pedal signal 1
J9-4	GND	Digital GND	J9-18	BRK_PP_AIN2	Brake pedal signal 2
J9-5	VCM_PPS_AIN	Vacuum pump pressure sensor	J9-19	BRK_PS_5V0	5 V supply for Brake pedal
J9-6	VCM_PPS_5V0	5 V supply for vacuum pump pressure sensor	J9-20	BRK_PS_5V0	5 V supply for Brake pedal
J9-7	GND	Digital GND	J9-21	GND	Digital GND
J9-8	LSD_REV_OUT_2	LSD output 8	J9-22	GND	Digital GND
J9-9	LSD_REV_OUT_3	LSD output 9	J9-23	DCDC_EN_DOUT	LSD output 0
J9-10	LSD_REV_OUT_4	LSD output 10	J9-24	WATER_PUMP_ RLY_DOUT	LSD output 1
J9-11	AC_OUTLET_ EM_RLY2_DOUT	LSD output 5	J9-25	AC_PWR_ RLY_DOUT	LSD output 2
J9-12	LSD_REV_OUT_0	LSD output 6	J9-26	AC_PTC_ RLY_DOUT	LSD output 3
J9-13	LSD_REV_OUT_1	LSD output 7	J9-27	AC_OUTLET_ EM_RLY1_DOUT	LSD output 4
J9-14	GND_RELAY	Analog GND	J9-28	GND_RELAY	Analog GND
J10-1	CAN0_H	CAN PHY CANH1	J10-21	CAN2_H	CAN PHY CANH
J10-2	CAN0_L	CAN PHY CANL1	J10-22	CAN2_L	CAN PHY CANL
J10-3	CAN1_H	CAN PHY CANH2	J10-23	GND	Digital GND
J10-4	CAN1_L	CAN PHY CANL2	J10-24	LIN_M	LIN PHY LIN1
J10-5	GND	Digital GND	J10-25	LIN_S	LIN PHY LIN2

Table 2. Board interface connector details*...continued*

Connector	Signal	Description	Connector	Signal	Description
J10-6	KEY_ON_DIN	MSDI SG0	J10-26	GND	Digital GND
J10-7	KEY_START_DIN	MSDI SG5	J10-27	BRAKE_SW_DIN	brake switch input
J10-8	AC_SW_DIN	MSDI SG1	J10-28	DRIVE_ GEAR_DIN	Driver gear switch input
J10-9	PTC_SW_DIN	MSDI SG2	J10-29	REVERSE_ GEAR_DIN	Reverse gear switch input
J10-10	CHARGE_CC_DIN	MSDI SG3	J10-30	PARK_GEAR_DIN	Parking gear switch input
J10-11	CRUISE_CTL_DIN	MSDI SG4	J10-31	NEUTRAL_ GEAR_DIN	Neutral gear switch input
J10-12	Reserved_AIN0	MSDI SP0	J10-32	HAND_ BRAKE_DIN	Hand brake switch input
J10-13	Reserved_AIN1	MSDI SP1	J10-33	Reserved_DIN0	MSDI SG6
J10-14	Reserved_AIN2	MSDI SP2	J10-34	Reserved_DIN1	MSDI SG7
J10-15	Reserved_AIN3	MSDI SP3	J10-35	Reserved_DIN2	MSDI SG8
J10-16	PWM_IN0	PWM input	J10-36	Reserved_DIN3	MSDI SG9
J10-17	PWM_IN1	PWM input	J10-37	GND	Digital GND
J10-18	GND	Digital GND	J10-38	CRUSH_SS_AIN	Crush sensor signal
J10-19	EPHY_TRX_P	Ethernet Differential signal (+)	J10-39	ALT_PP_AIN1	Accelerator pedal signal 1
J10-20	EPHY_TRX_N	Ethernet Differential signal (-)	J10-40	ALT_PP_AIN2	Accelerator pedal signal 2
J11-1	GND_RELAY	Analog GND	J11-9	GND_RELAY	Analog GND
J11-2	GND_RELAY	Analog GND	J11-10	GND_RELAY	Analog GND
J11-3	PRE_CHARGE_ DOUT	HSD output 3	J11-11	$MAIN_+$_ DRV_DOUT	HSD output 1
J11-4	COOL_FAN_ DRV_DOUT	HSD output 4	J11-12	HEAT_DRV_DOUT	HSD output 2
J11-5	GND	Digital GND	J11-13	KL_WAKE_DIN	SBC WAKE1
J11-6	VBAT_12V	Power supply	J11-14	VBAT_12V	Power supply
J11-7	VBAT_12V	Power supply	J11-15	VBAT_12V	Power supply
J11-8	VBAT_12V	Power supply	J11-16	VBAT_12V	Power supply

The following table lists the interface definition for OBC application.

Table 3.

Function	Connector	Signal	Description
PFC	OBC_PFC_PWMA	PWM_0_A[2]	High-frequency
	OBC_PFC_PWMB	PWM_0_B[2]	
	OBC_PFCREC_PWMA	PWM_0_A[3]	Low-frequency
	OBC_PFCREC_PWMB	PWM_0_B[3]	

Table 3. *...continued*

Function	Connector	Signal	Description
	OBC_PFC_PHASE_ADC	ADC1_P0	PFC phase voltage
	OBC_PFC_BUS_ADC	ADC1_P1	PFC output voltage
	OBC_PFC_CINPUT_ADC	ADC1_P2	PFC input current
	OBC_PFC_CTH_ADC	ADC5_P0	PFC upper-MOS drain current
	OBC_PFC_CTL_ADC	ADC5_P6	PFC lower-MOS drain current
	OBC_PFC_CTH_FAULT	eFlexPWM_0_Fault3	PFC input current comparator
	OBC_PFC_CTL_FAULT	eFlexPWM_0_Fault2	PFC input current comparator
	OBC_REL_GPIO	GPIO[2]	Precharge
LLC	OBC_LLC_PWMA0	PWM_1_A[0]	Transformer primary side
	OBC_LLC_PWMB0	PWM_1_B[0]	
	OBC_LLC_PWMA1	PWM_1_A[1]	
	OBC_LLC_PWMB1	PWM_1_B[1]	
	OBC_LLC_PWMA2	PWM_1_A[2]	Transformer secondary side
	OBC_LLC_PWMB2	PWM_1_B[2]	
	OBC_LLC_PWMA3	PWM_1_A[3]	
	OBC_LLC_PWMB3	PWM_1_B[3]	
	OBC_LLC_CT_FAULT	eFlexPWM_1_Fault3	CLLLC input current comparator
	OBC_LLC_IOUT_ADC	ADC2_P0	CLLLC output current
	OBC_LLC_IPROT_ADC	ADC2_P1	CLLLC output voltage
	OBC_LLC_IPRI_ADC	ADC2_P3	CLLLC primary CT
	OBC_LLC_V_OUT	ADC4_P2	CLLLC primary CT filter
Nonfunctional	OBC_TEMP2_SENSE	ADC0_P0	Temperature 2 detection
	OBC_TEMP1_SENSE	ADC0_P1	Temperature 1 detection

3 Software user guide

BMS and VCU software is based on AUTOSAR drivers, including:

- SW32K3_RTD
- MC33774 BCC
- MC33665 TPL PHY
- SBC

It provides a series of AUTOSAR-specific API to facilitate the BMS application designer to implement the BMS application.

3.1 File structure

The following table shows the structure and content of the project folder.

Table 4. MCAL Project Files

Folders/Files	Descriptions
HSE_LIB_0_2_7_0	Contains HSE application-related interface files.
S32K376_BMSVCU	BMS main folder contains the project of each core and OTA debug tools.
S32K396_BootLoader	Bootloader main folder contains bootloader project and debug tools.
README.md	About application description, issue, revision history, and I/O usage.

3.2 Software block diagram

Figure 24 shows the S32k376 BMS and VCU solution software block diagram. BMS function runs on function on core CM7_0, VCU function runs on lockstep on core CM7_2 and monitors another two cores through IPCF. OTA and reserved OBC functions run on the function core CM7_1. Bootloader and OTA for updating firmware. The GUI displays BMS and VCU information. All drivers is based on RTD.

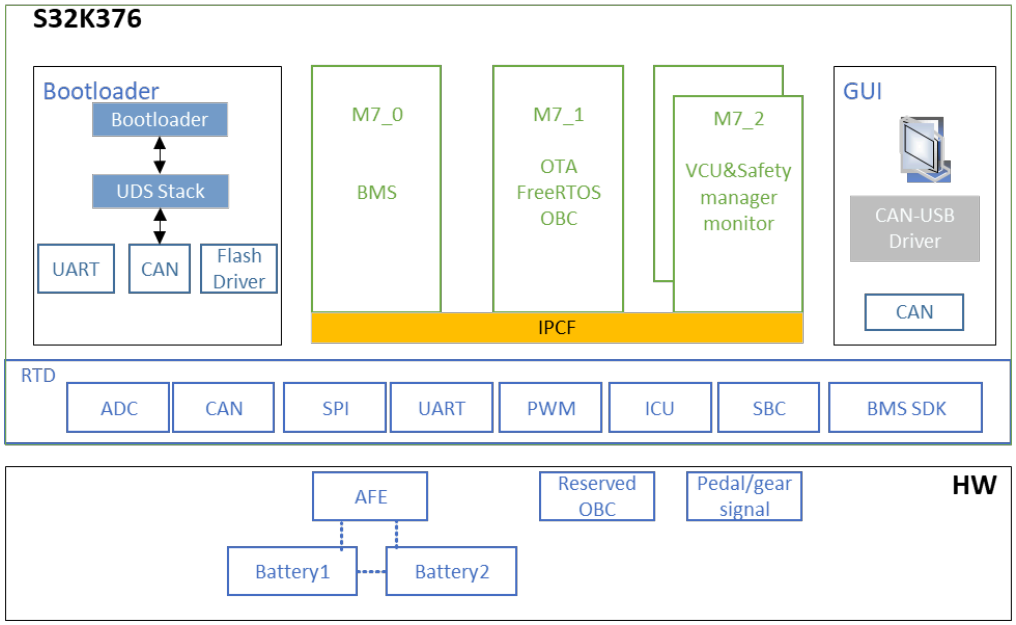


Figure 24. BMS and VCU software block diagram

3.3 Basic workflow of a project

The BMS and VCU reference design is a three core architecture. The core0 and core1 are functional cores. The core2 is the safety lockstep core. BMS, VCU, and OTA are running on different cores. Core0 is the startup core. It brings up first, then core1 and core2. The core0 initial system and peripheral clock and setup GPIO status. Using the CM7-x-ENABLE field of IVT boot configuration word to disable core1 and core2 to avoid core1 and core2 executing uninitialized peripheral. The core1 and core2 startup after core0 write the correct address to the MC_ME module.

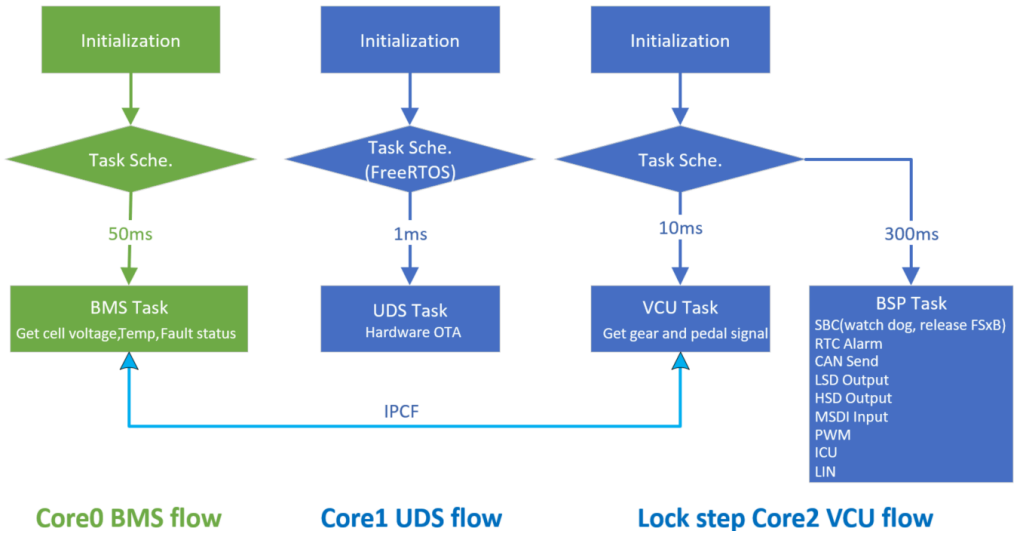


Figure 25. MCAL demo project programming flow diagram

- BMS tasks run on Core0, scheduled by general timer.
- OTA tasks run on Core1, scheduled by FreeRTOS.
- VCU and BSP tasks, run on Core2, scheduled by general timer.
- IPCF is used for multicore communication.
- Start up the 1st and the 2nd core after starting zero core by setting IVT and MC_ME.
- OTA and reserved OBC function on the function core CM7_1
- Bootloader and OTA for updating firmware
- GUI displays BMS and VCU information

3.4 BMS program flow

BMS task initializes MC33665A and MC33774A, then get each BCC voltage and faults state by the default dual SPI mode interface. The brief BMS program flow is show in Figure 26, there is a state machine in the running state. A few safety mechanisms have been implemented on this state machine for directing the customer how to enable the BMS safety library.

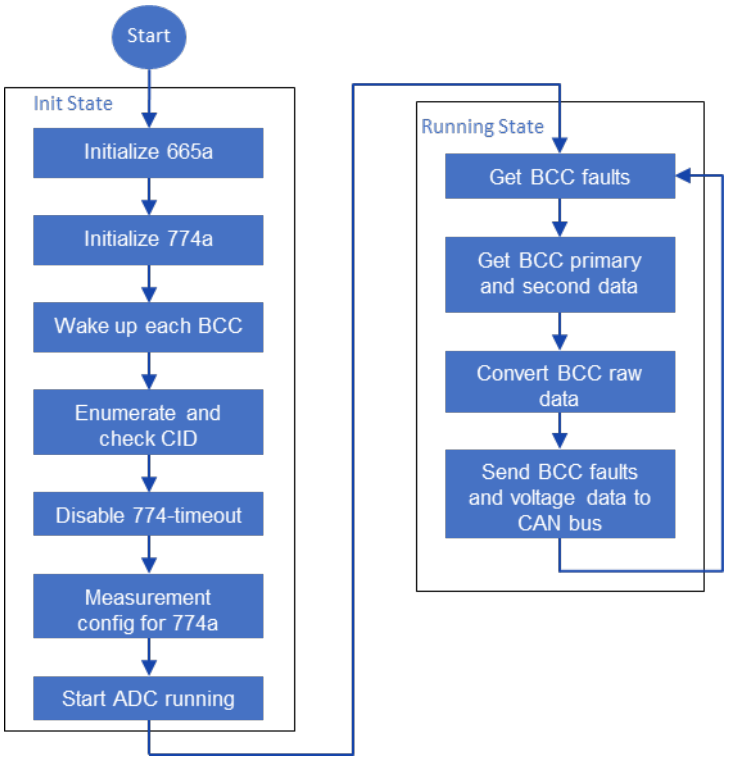


Figure 26. BMS program flow

Regarding the running state machine, there are five major cycles. Each cycle has ten minor cycles. The major cycle increase one after the minor cycle increase to ten cycles.

3.5 SAF function

S32K396 Safety Software Framework (S32K396 SAF) is a software product that contains software components establishing the safety foundation for customer safety applications compliant with ISO 26262 functional safety. S32K396 SAF is a framework that implements the device safety concept system solution. See ***S32K396_SAF_Safety_Manual.pdf*** and package **S32K396_SAF_0.8.0** for more information on SAF, because the SAF is not free for the customers.

S32k376 BMS&VCU solution enabled Figure 27 highlighted component. However, customers can only use it if they purchase the SAF installation package. For the detailed setup steps, please refer to QSG. The BIST manager, Extended microcontroller Error manager, and sBoot are executed in the initialization phase, and SquareCheck is executed in the runtime phase.

Component	Fully Controlled IPs	Partially Controlled IPs
BIST Manager	STCU2, SEFTEST_GPR	MC_RGM, DCM_GPR
Extended Microcontroller Error Manager	FCCU, ERM, EIM	DCM, XBIC
Mode Selector	-	-
sBoot	-	EDMA, SIUL2, CMU, MC_CGM, SWT, STM, FLEXCAN, PIT, SPI, FCCU, XRDC, SCB, MC_RGM, EMAC
SquareCheck	-	CMU, Cores (CM7_0, CM7_1, CM7_2), CRC, EDMA, EIM, ERM, EMAC, HSE, MC_CGM, MC_RGM, MSCM, SRAMC, STM, SWT, XRDC
Software Recovery	-	MC_ME

Figure 27. S32K396 SAF components

3.6 BMS SL function

This is the S32K3 BMS SL SDK DEMO AUTOSAR 4.4 Version 1.0.0 release for the S32K3 platform. The Bcc_775a_SL, Bcc_774a_SL, Bcc_772c_SL, and Bms_TPL3_SL_E2E libraries included in this release have DEMO quality status in terms of testing and quality documentation. DEMO libraries are not qualified and contain a partial feature set. It is not intended to be used in production. This DEMO version of the Safety Library is delivered in binary format, and it's intended for evaluation purposes only.

See ***RTD_BCC_774A_SL_UM.pdf*** and package **SW32K3_BMS_SL_SDK_4.4_1.0.0_DEMO** for more information on BMS SL because it is not free for customers.

S32k376 BMS&VCU solution enabled external safety mechanisms 1, 2, 3, and 5 to monitor battery status. Customers can only use it if they purchase the BMS SL installation package. For the detailed setup steps, see QSG.

3.7 Project memory map

S32K376 have 6M flash space to store code binary that consists of three 2M flash blocks. The bootloader is placed in the first flash block, and the three core projects are divided into one flash block via link file and MPU. OTA for code-switching in block 1 and block 2.

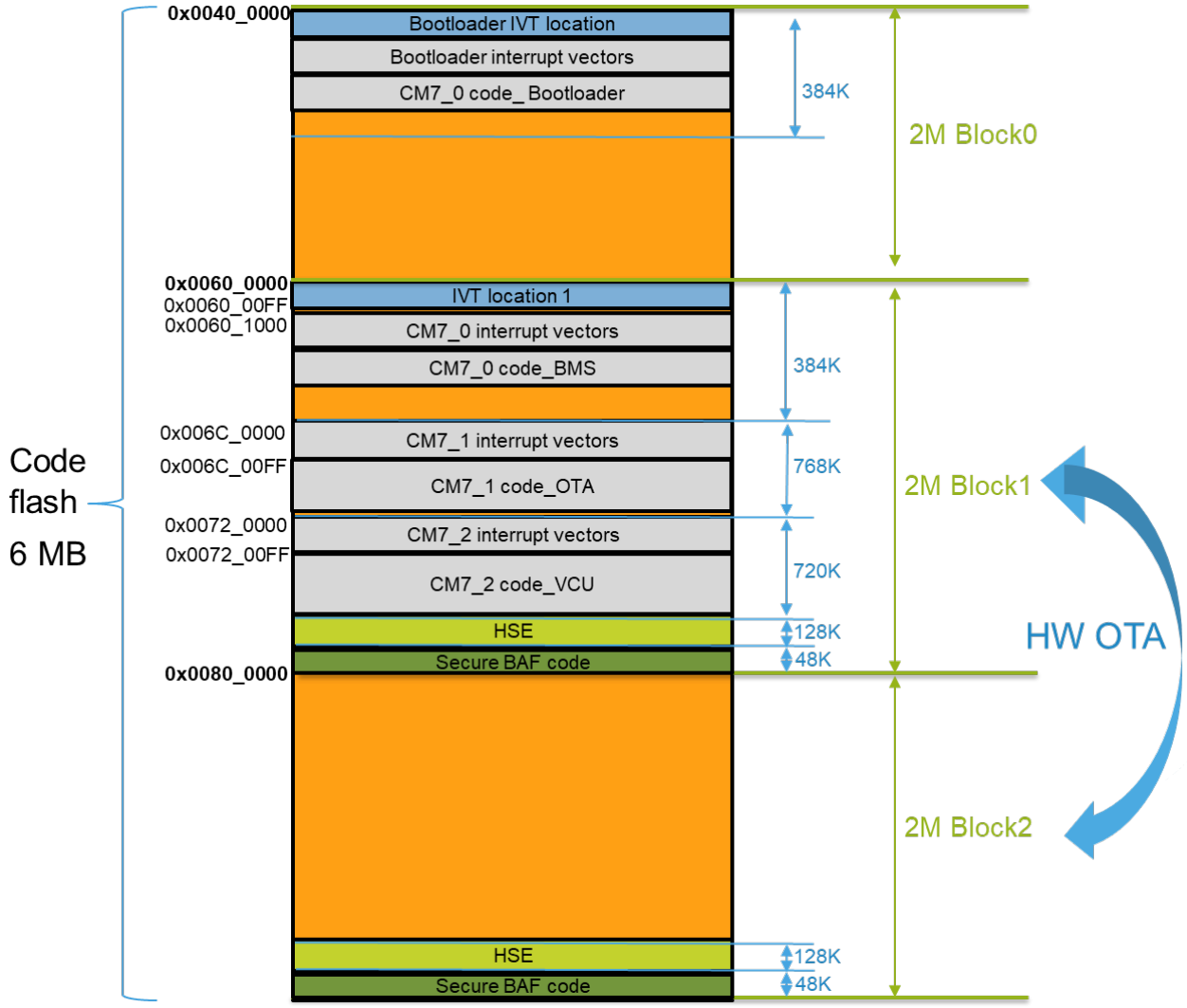


Figure 28. Memory map

3.8 Software and tools required

- Configuration tool: EB Tresos Studio 29.0.0.
- MCAL drivers: SW32K3_RTD_4.4_R21-11_3.0.0_P01.
- BMS CDD: SWS32K3_BMS_SDK_4.4_1.0.0
- Compiler: GCC version 10.2.0 20200723.
- Debugger: Lauterbach TRACE32 JTAG Debugger.
- CAN adapter: PCAN-USB Pro.

Figure 29. Lauterbach debugger

Figure 30. PCAN-USB Pro

3.9 How to build

1. Open EB Tresos Studio and import EB Tresos project. There are three cores project to split-up CM7_0, CM7_1, and lockstep core CM7_2.

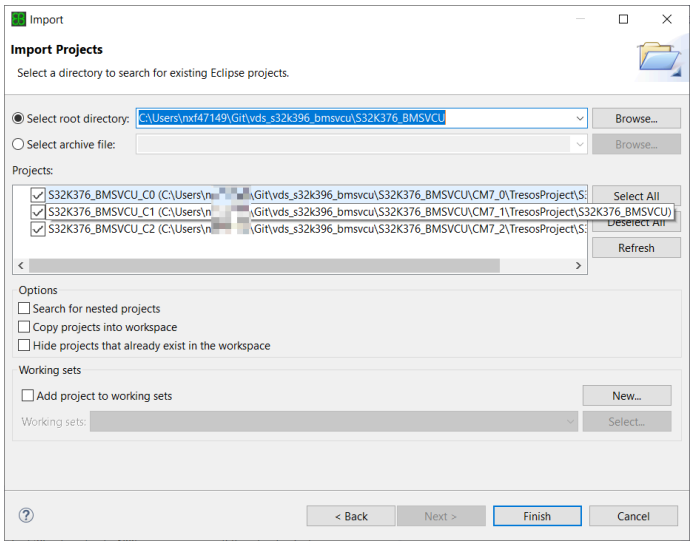


Figure 31. Import project

2. Double-click the project name “S32K376_BMSVCU_C0” and click the Generate button. The other two projects must perform the same operation.

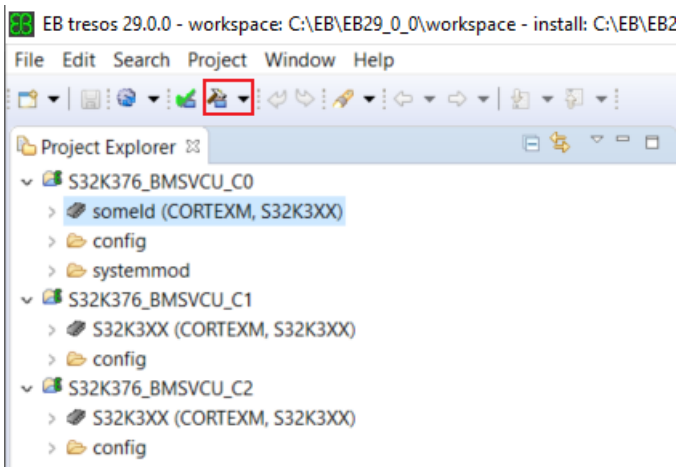


Figure 32. Generation of source file

3. Fill in the user-defined paths for compilation in project_parameters.mk, at each project. Normally, the user must only modify the GCC_DIR path to the user's location; other paths have been set to a relative path. The other two projects must perform the same operation.

```
#The path to the GCC installation dir
GCC_DIR = C:/NXP/S32DS.3.4/S32DS/build_tools/gcc_v10.2/gcc-10.2-arm32-eabi
```

Figure 33. Compile configuration

4. Change the current directory to S32K376_BMSVCU/CM7_0, and execute "make build -j8" (the-j8 option depends on the performance of the user's personal computer). The other two projects must perform the same operation.

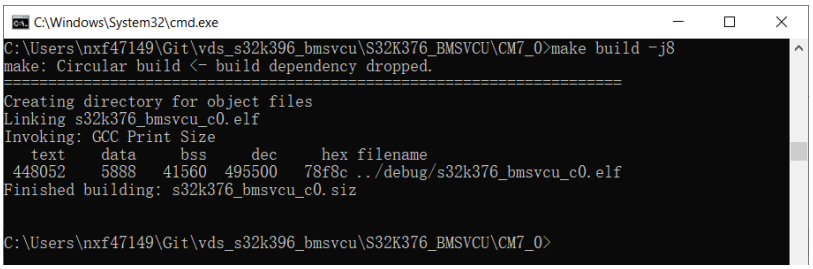


Figure 34. Compilation information

5. You can execute "make clean to clear output objects" in unexpected situations and build again.

3.10 How to download.elf file

1. Open TRACE32 and select File->Run Script…

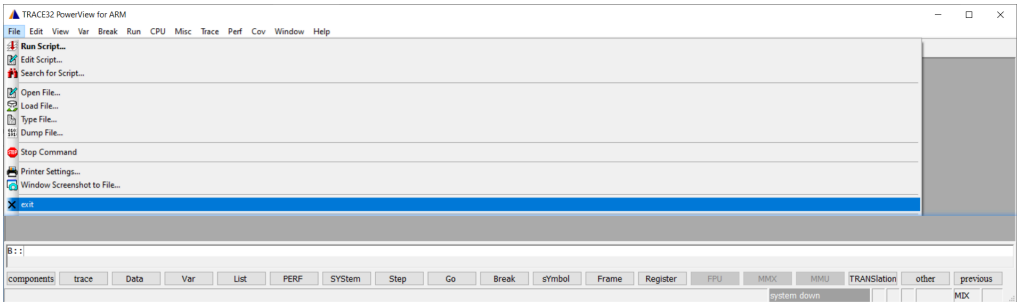


Figure 35. TRACE32 homepage

2. Select S32K376_flash_triple_core012.cmm (in S32K376_BMSVCU/debug directory) to download all 3x project.elf files into flash.
3. When the debugger finishes loading.elf into flash, it pops out three debug windows corresponding to each project. Click the Go button to run.

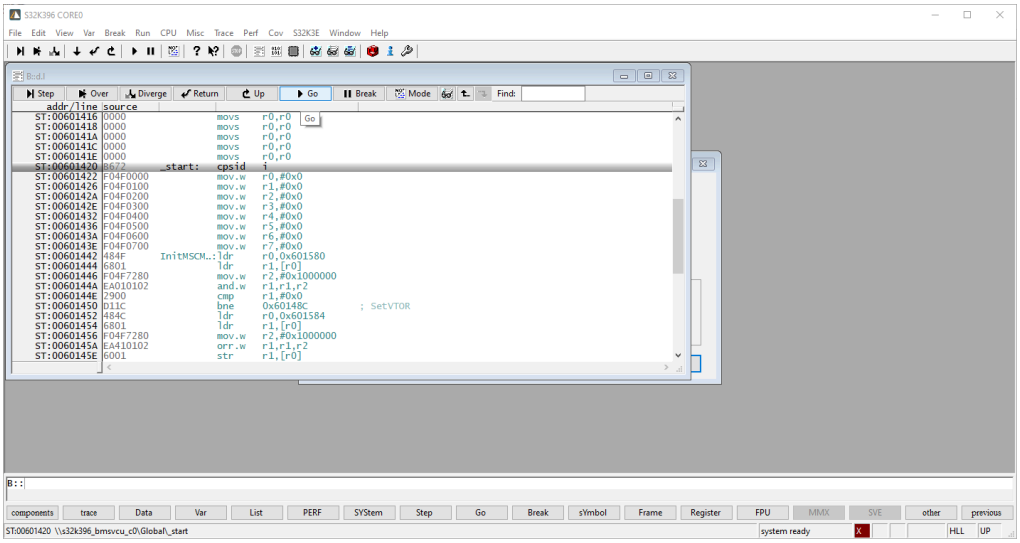


Figure 36. Debug window

Note: *When you download the BMS and VCU project and ignore the bootloader, please confirm that there is no IVT on flash block 0. Otherwise, the BMS and VCU projects can't startup normally. If IVT is in block 0, open the following comments in S32k396_flash_triple_core012.cmm to erase block 0 and reload the BMS&VCU project.*

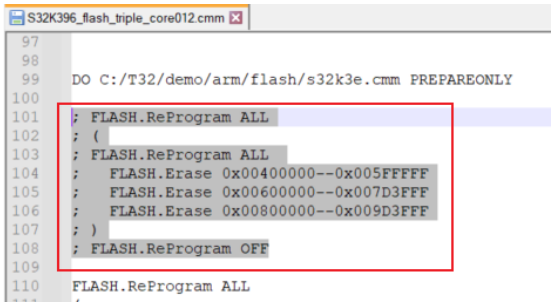


Figure 37. The script to erase the block

4 OTA

4.1 Overview

The project-integrated OTA in Core 1 and is scheduled by FreeRTOS. For users of the S32K3xx device with OTA enabled, the passive region can be read and written, and when an OTA update is performed, the new APP is always written to the passive region, but no program can execute in that region. OTA code flash memory layout and swap are shown in Figure 25.

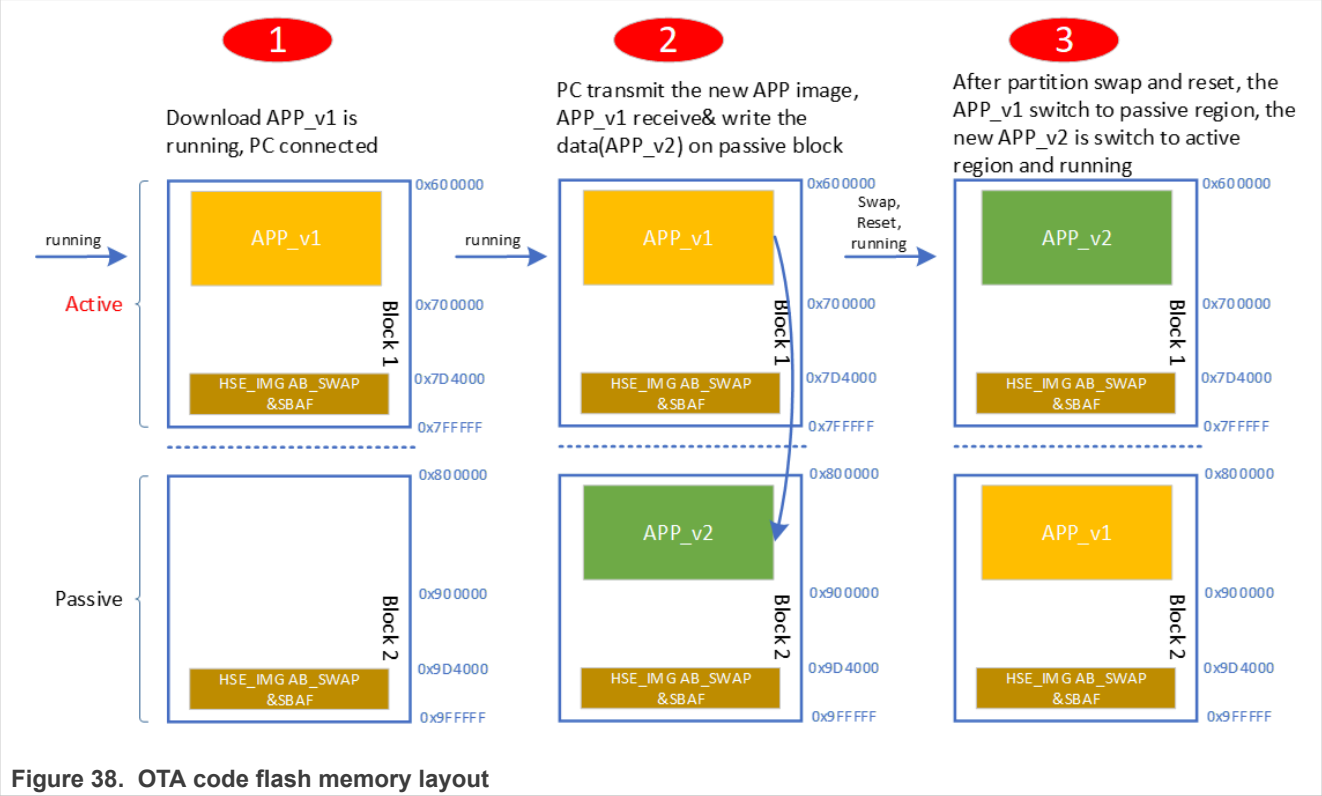


Figure 38. OTA code flash memory layout

1. The OTA update agent-APP_V1 is first downloaded to the active block1 (0x00600000) via the debugger and then starts running.
2. When the OTA update is performed, PC sends the new version of the OTA update agent-APP_V2, APP_V1 receives the binary file of APP_V2, and writes it to the block2(0x0800000), which is in passive.
3. After partition swap and reset, the APP_V1 switch to the passive region, and the new APP_V2 is switched to the active region and running.

4.2 OTA memory map

If you install HSE A/B Swap FW to MCU, the 176Kb end of code flash block1 and block2 become Secure NVM after one reset. HSE and Secure BAF use the Secure NVM. So, the actual available flash space is 1872 kB. Figure 23 shows a detailed memory map.

4.3 How to debug

1. Debug preparation
   - MCU have installed HSE A/B swap FW. Customer can download HSE firmware on the official website and step by step refer ***HSE_DEMOAPP_S32K3XX_0_2_7_0_ReadMe.pdf*** to install HSE firmware.
   - The ECUBus0.2.24 installation package can be found in the following Figure 39.

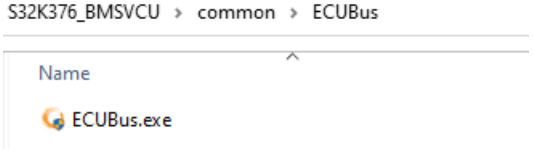


Figure 39. ECUBus installation package

- Connect BMU through Ethernet, and modify the IP address of the PC to an unused address in the 192.168.0.x network segment showing in the blew Figure 40

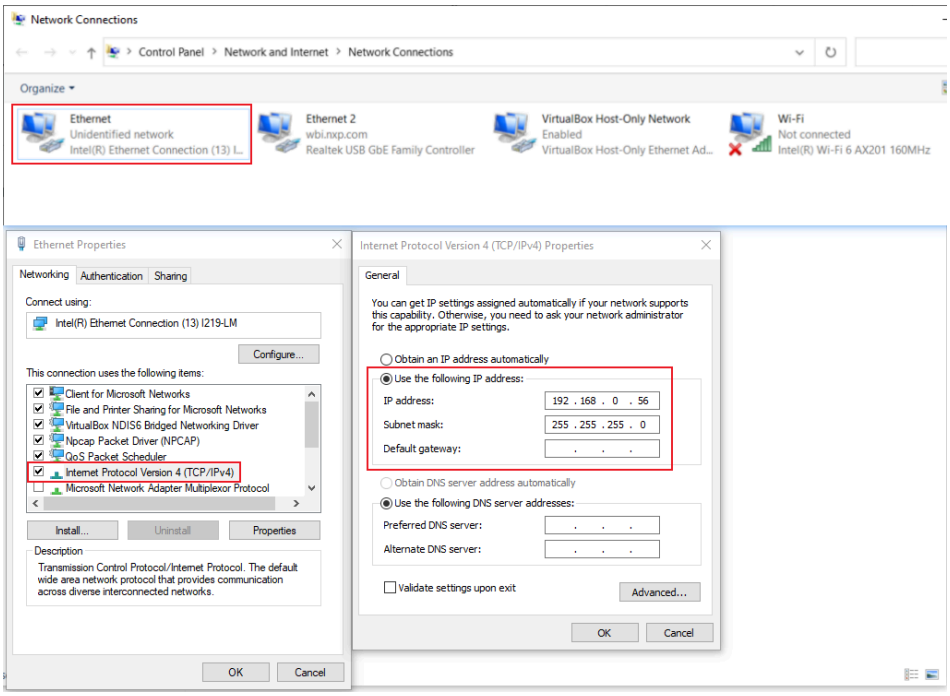


Figure 40. IP address Configuration of PC

2. Download BMS&VCU application into flash memory.
3. The application running normally.
4. Open ECUBus- Click Powerful UDS tester "Go" button.

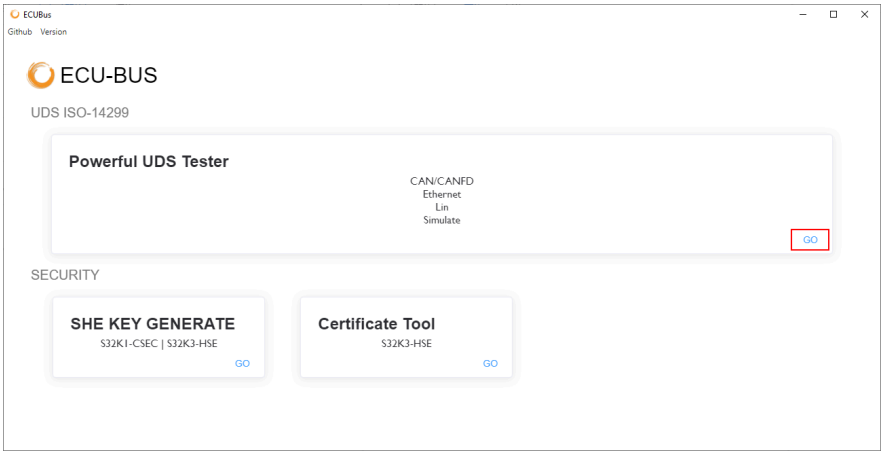


Figure 41. ECUBus homepage

5. Open UDS project- Choose the UDS project folder by clicking Open UDS Project, (S32K376_BMSVCU\S32K396_ota_update_agent_scripts\OTA_update_agent). Then click the DoIP ISO-13400-2 "Go" button.

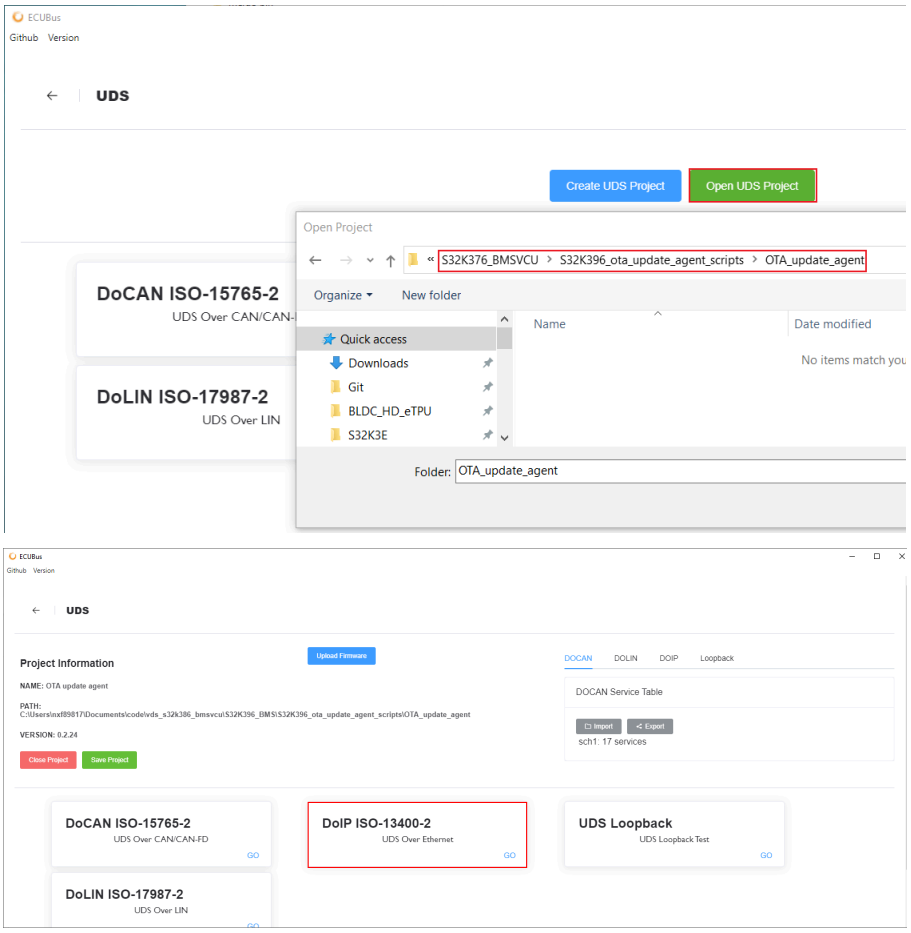


Figure 42. ECUBus Project folder

6. Configuration of Ethernet communication.
   - Click the settings button in the upper right corner.
   - Input **Multicast** as **192.168.0.255**, **SA** (source address) as **55**.
   - Click Search Device, then click Connect.

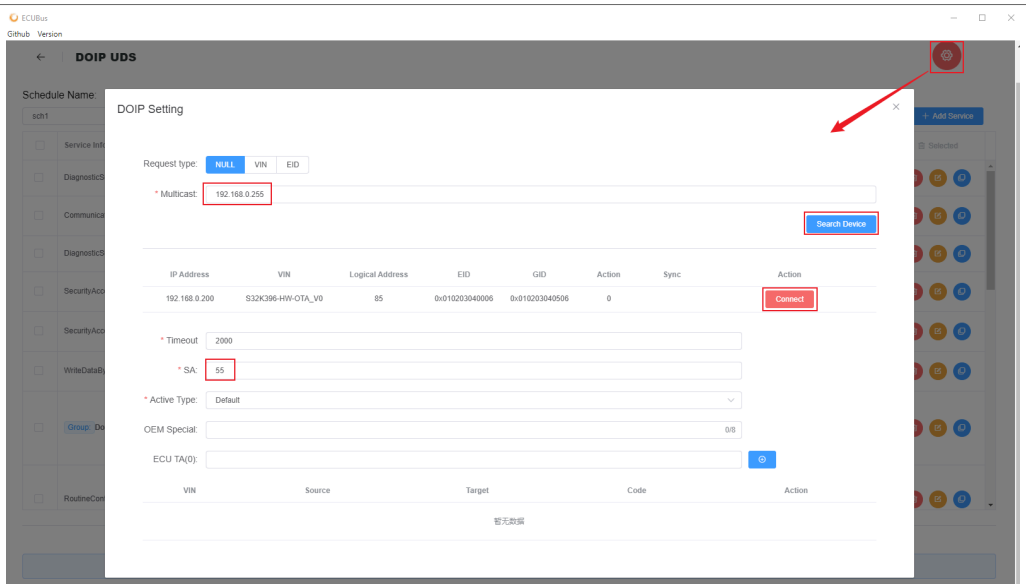
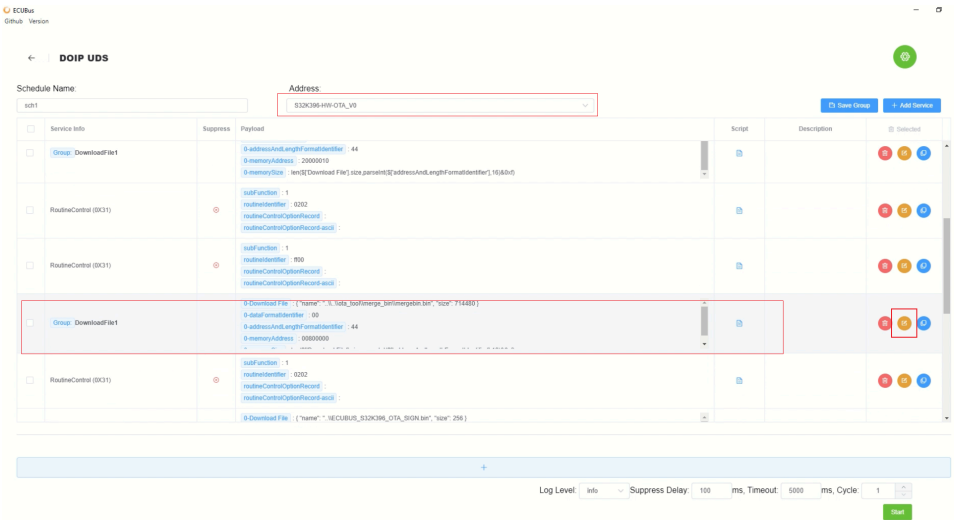


Figure 43. Configuration parameter of Ethernet

7. Preparation of update binary. Change the directory to S32K376_BMSVCU/ota_tool/merge_bin. Execute mergebin_ota.bat. It merges three project binaries into one binary. Select Address as S32K396-HW-OTA_V0. Then change the ECUBus service info 10th item, click the "Choose File" button, select the previously merged binary, and click the "Change Group" button.



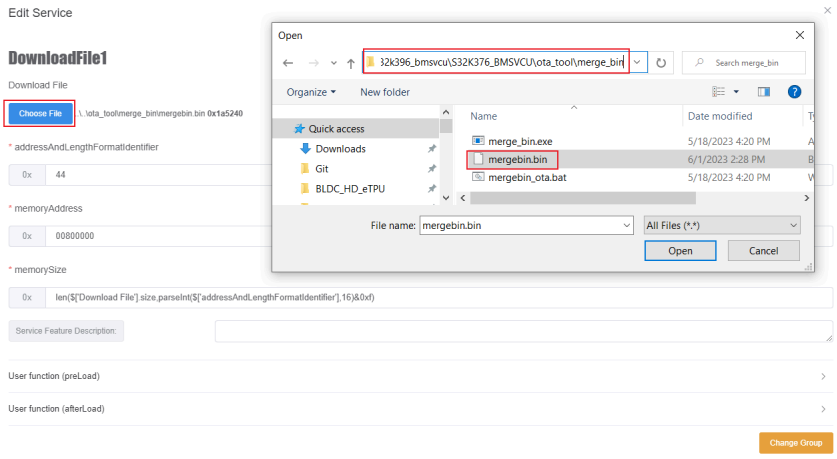


Figure 44. Configuration of the download file

8. Execute the OTA procedure.
   - Click the start button.
   - Wait for the binary to be downloaded successfully, and the code automatically runs after the download is complete.

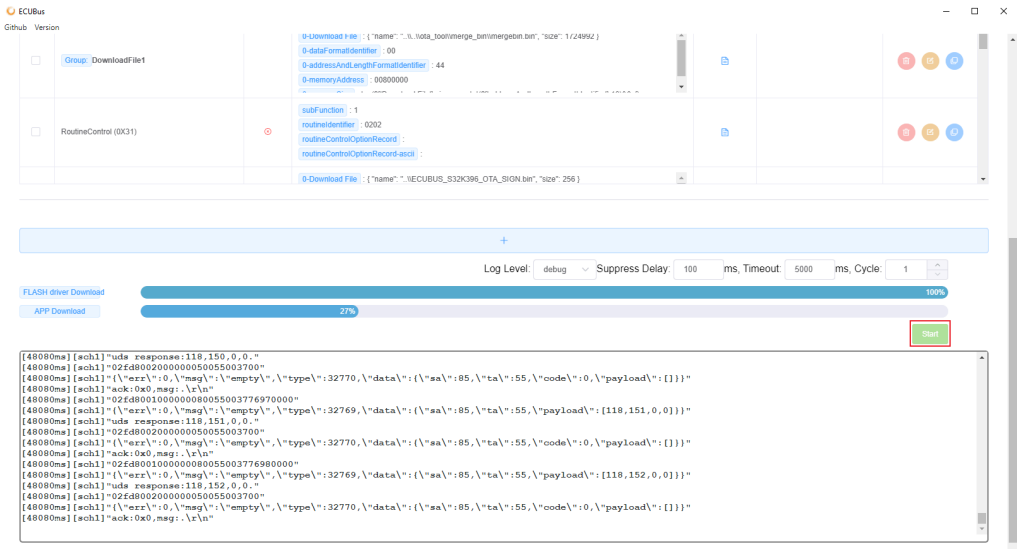


Figure 45. Interface of downloading binary

5 Bootloader user guide

5.1 Overview

A bootloader is provided to update the application via CAN bus if there are no tools to download the application directly. S32k396_Bootloader is a standalone project, which flashes on the flash block0. Figure 28 shows the IVT information that appears on the low address, so the code always starts up from block0. The application is downloaded on flash block1 by the bootloader.

There are two modes to enter bootloader mode:

- Determine whether to enter bootloader mode by verifying valid APP info. If the APP info is invalid, enter bootloader mode.

- Determine whether to enter bootloader mode by detecting the J10_27 and J10_32 status. If both J10_27 and J10_32 are 12 V high voltage, enter bootloader mode.

5.2 How to debug

1. Debug preparation.
   - ECUBus tool. Find the ECUBus installation package in Figure 39.
   - Connect BMU through CANFD.
2. Download the application into flash memory. Download s32k396_bootloader.elf into flash by Lauterbach with s32k396_flash_c0.cmm. APP is erased every time the script runs. Erasing command can be found in Figure 37.
3. The application is running normally.
4. Open ECUBus. Click the Powerful UDS Tester Go button.

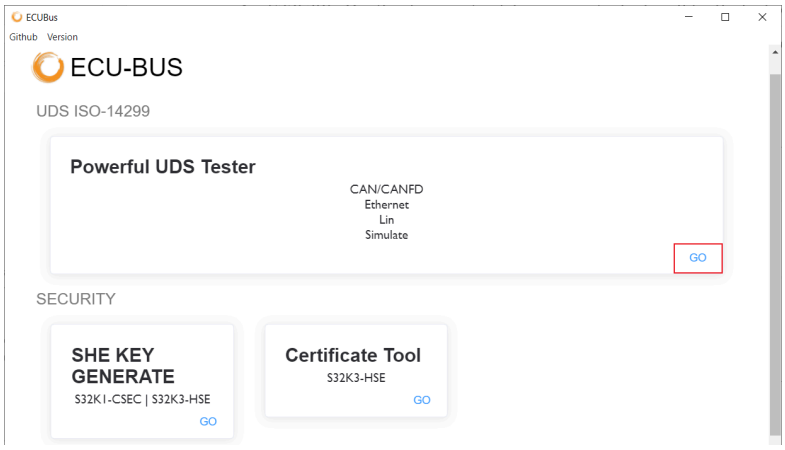
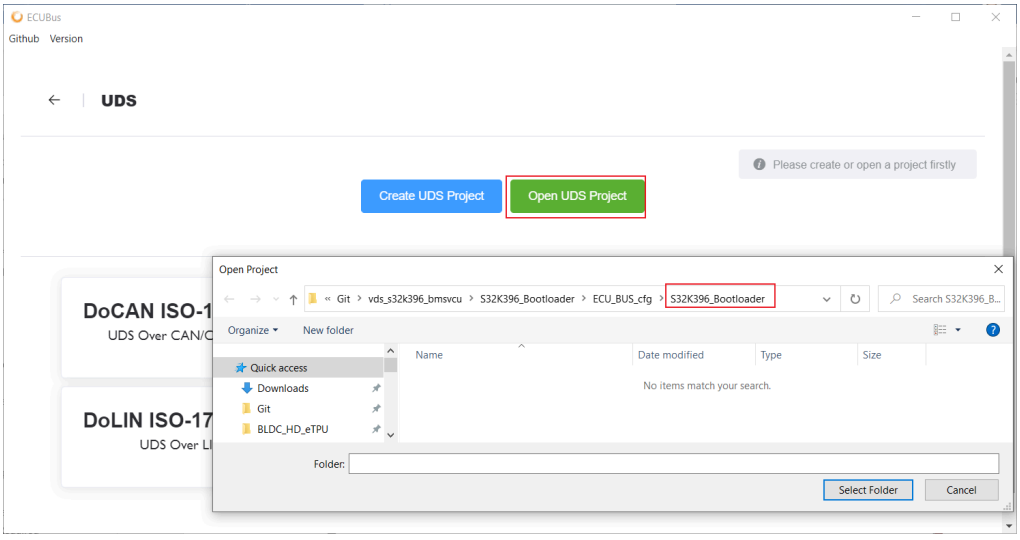


Figure 46. ECUBus homepage

5. Open UDS project. Choose the UDS project folder by clicking **Open UDS Project** (..\S32K396_Bootloader \ECU_BUS_cfg\S32K396_Bootloader). Then click the DoCAN ISO-15765-2 **Go** button.



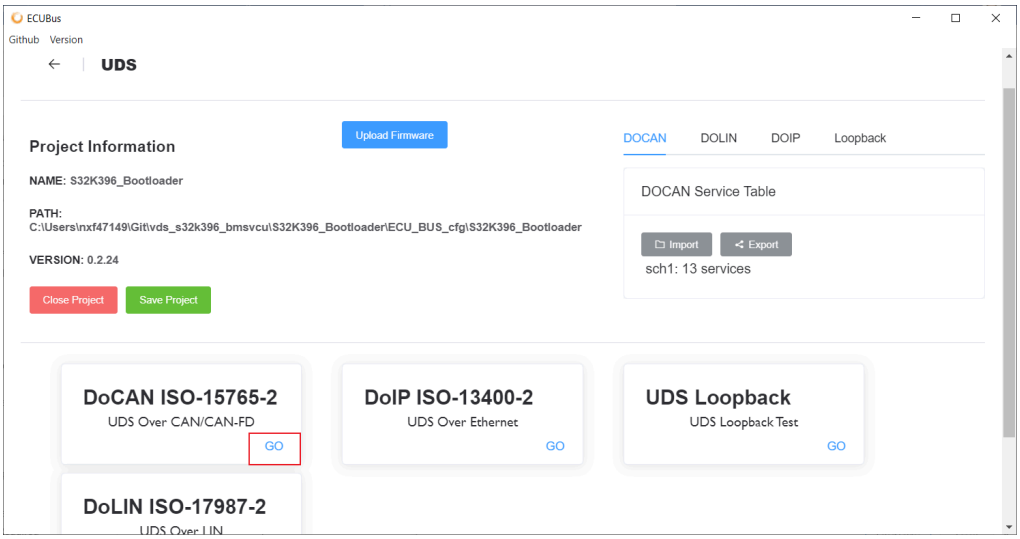
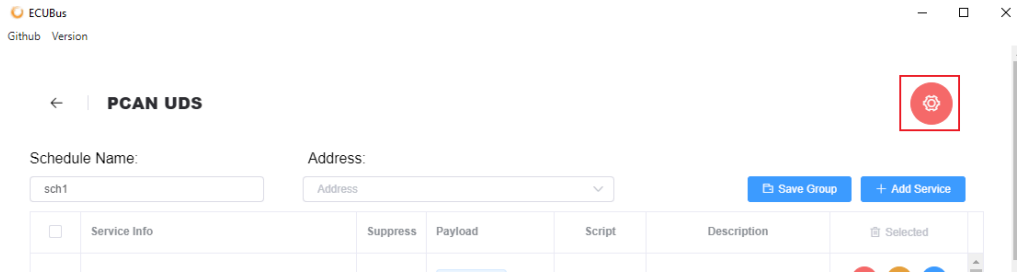


Figure 47. ECUBus UDS project folder

6. Configuration of CANFD communication.
   - Click the settings button in the upper right corner.
   - CAN-FD: Select the device, then check CAN-FD and check PADDING. Select the nominal bit rate as 500 kbit/s, the data bit rate as 2 Mbit/s, and TLC as 64. The following picture shows the CANFD parameters. Fill in values like below and click the **Connect** button.
   - Select Format as Normal fixed addressing, Address as Physical, and Message as Local.
   - Input SA (source address) as 35 and TA (target address) as 55.
   - Click ADD.



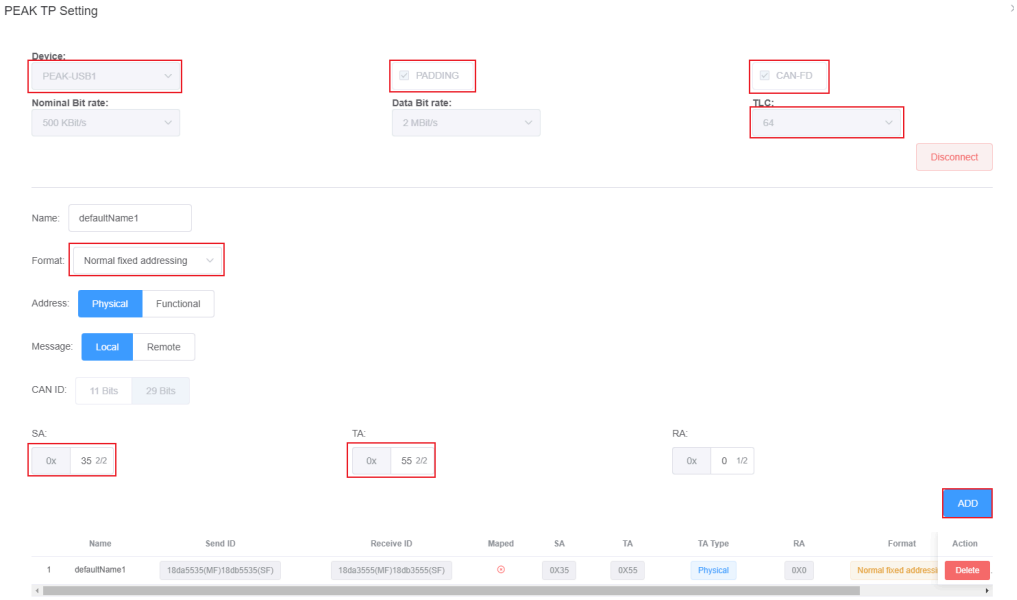
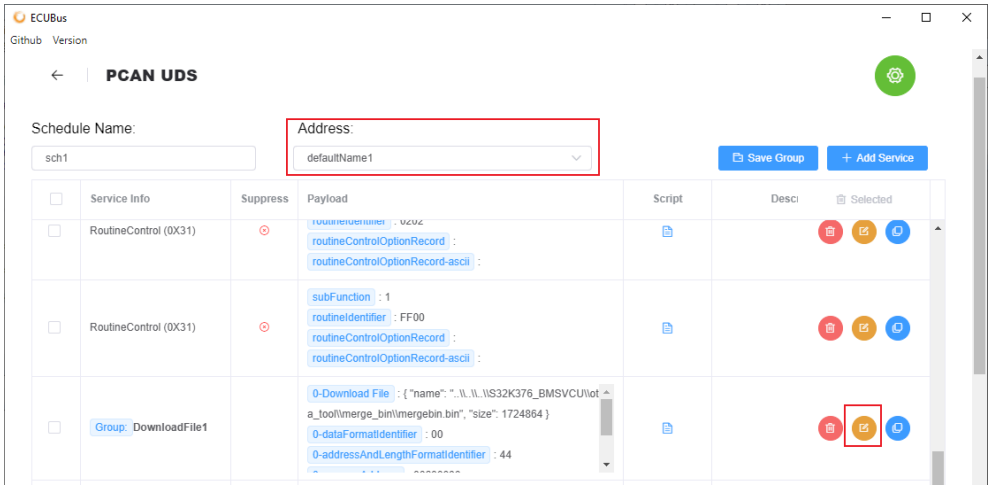


Figure 48. Configuration parameter of CANFD

7. Preparation of update binary.
   - Change the directory to S32K376_BMSVCU/ota_tool/merge_bin. Execute mergebin_ota.bat. It merges three project binaries into one binary.
   - Checked defaultName1 in the address box.
   - Then change the ECUBus service info 10th item, click the Choose File button and select the previously merged binary. Click the Change Group button.



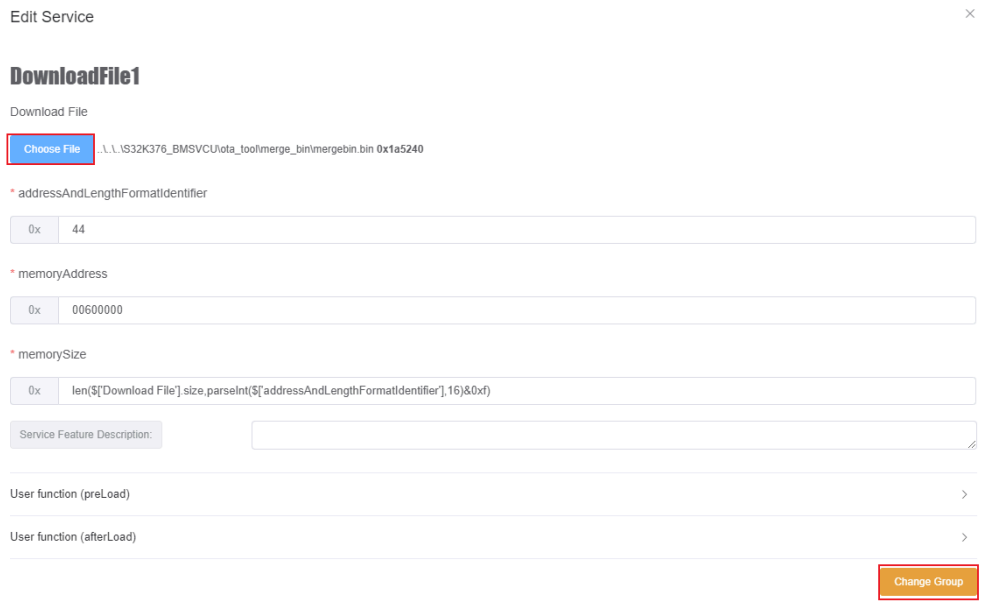


Figure 49. Configuration of the download file

8. Execute the OTA procedure.
   - Click the start button.
   - Wait for the binary to be downloaded successfully.
   - Press SW1 to reset BMU, the APP has installed in the flash bock1, and the code always jumps to block1 to execute the APP when the code starts.

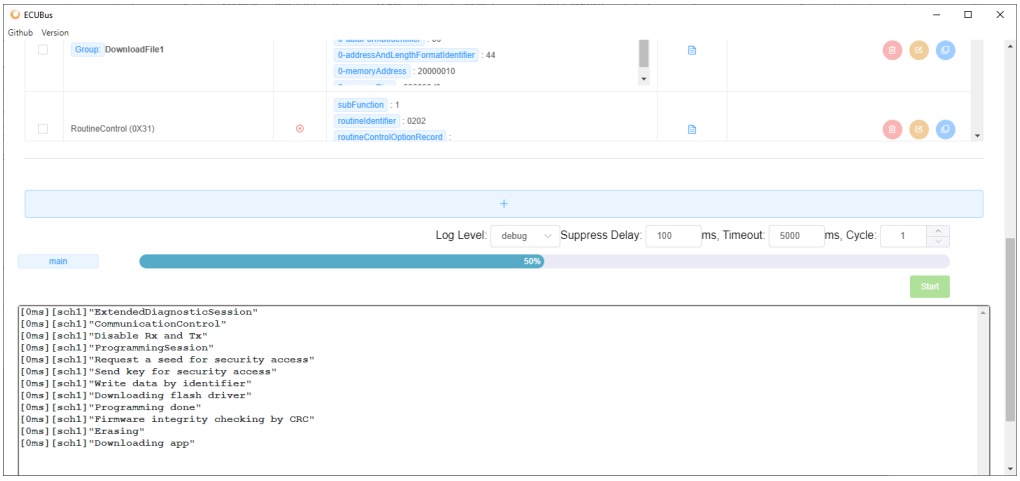


Figure 50. Interface of downloading binary

6 GUI user guide

6.1 GUI interface introduction

This GUI displays data transmitted from BMS and VCU systems via CAN bus. Hands-on instructions are listed below.

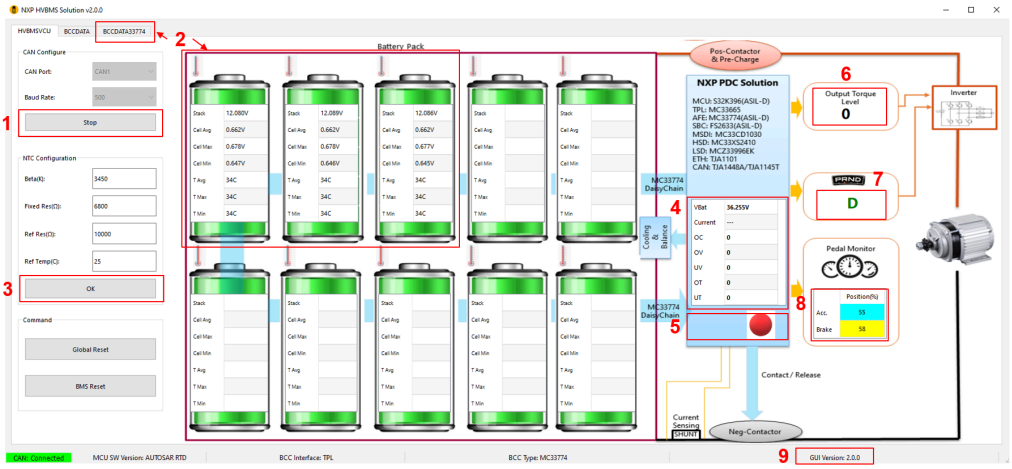
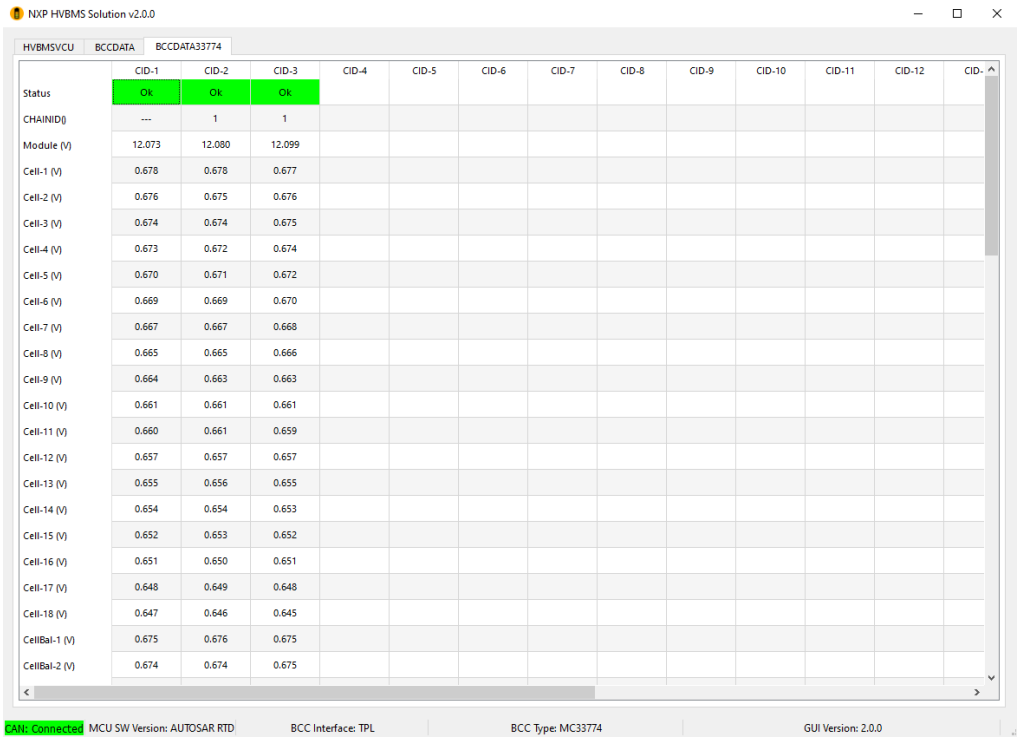


Figure 51. GUI layout Info

1. Select the CAN port and baud rate (500 kbit/s for this demo) of CAN transmission then click the Start button. Click the Stop button when you want to disconnect the CAN adapter.
2. The battery voltage is displayed, including stack voltage and cell voltage(average, maximal, and minimal). Temperature(average, maximal, and minimal) is displayed too. This GUI supports up to ten battery stacks.
3. NTC parameters can be changed according to the populated NTC.
4. Total voltage, current (not measured in this demo), and fault information is displayed.
5. This indicator becomes green when the VCU platform is started; otherwise becomes red.
6. Output torque level (0, Low, Mid, and High) is shown according to the accelerated pedal range.
7. Gear position (R, N, and D) is shown.
8. Positions of the accelerate pedal and brake pedal is shown. If the measured values are out of range, an error image appears.
9. Basic system information is shown on the status bar.

All BCC data are displayed on the BCCDATA page, see Figure 52, where the first two rows indicate the CID's status, including OK, error number, and so on.



	CID-1	CID-2	CID-3	CID-4	CID-5	CID-6	CID-7	CID-8	CID-9	CID-10	CID-11	CID-12	CID-
Status	Ok	Ok	Ok										
CHAINID()	---	1	1										
Module (V)	12.073	12.080	12.099										
Cell-1 (V)	0.678	0.678	0.677										
Cell-2 (V)	0.676	0.675	0.676										
Cell-3 (V)	0.674	0.674	0.675										
Cell-4 (V)	0.673	0.672	0.674										
Cell-5 (V)	0.670	0.671	0.672										
Cell-6 (V)	0.669	0.669	0.670										
Cell-7 (V)	0.667	0.667	0.668										
Cell-8 (V)	0.665	0.665	0.666										
Cell-9 (V)	0.664	0.663	0.663										
Cell-10 (V)	0.661	0.661	0.661										
Cell-11 (V)	0.660	0.661	0.659										
Cell-12 (V)	0.657	0.657	0.657										
Cell-13 (V)	0.655	0.656	0.655										
Cell-14 (V)	0.654	0.654	0.653										
Cell-15 (V)	0.652	0.653	0.652										
Cell-16 (V)	0.651	0.650	0.651										
Cell-17 (V)	0.648	0.649	0.648										
Cell-18 (V)	0.647	0.646	0.645										
CellBal-1 (V)	0.675	0.676	0.675										
CellBal-2 (V)	0.674	0.674	0.675										

Figure 52. BCC data table page

6.2 GUI data CAN protocol

The following figure lists all the protocols that interact data between GUI and BMU. The ID 0x1881xxxx to 0x1887xxxx are used for BMS data, and the ID 0x1889xxxx is used for VCU communication. Customers can parse those data on either PCAN viewer.

BCC Measurement

CAN ID [Hex]	Byte[0:1]	Byte[2:3]	Byte[4:5]	Byte[6:7]
	Stack voltage measurement	Cell14 voltage measurement	Cell13 voltage measurement	Cell12 voltage measurement
0x18810x04	Cell11 voltage measurement	Cell10 voltage measurement	Cell9 voltage measurement	Cell8 voltage measurement
0x18810x08	Cell7 voltage measurement	Cell6 voltage measurement	Cell5 voltage measurement	Cell4 voltage measurement
0x18810x0C	Cell3 voltage measurement	Cell2 voltage measurement	Cell1 voltage measurement	AN6 voltage measurement
0x18810x10	AN5 voltage measurement	AN4 voltage measurement	AN3 voltage measurement	AN2 voltage measurement
0x18810x14	AN1 voltage measurement	AN0 voltage measurement	IC temperature measurement	AN7 voltage measurement
0x18810x20	Cell15 voltage measurement	Cell16 voltage measurement	Cell17 voltage measurement	Cell18 voltage measurement
0x18810xA0	VBAT voltage measurement	Cell14 balance voltage	Cell13 balance voltage	Cell12 balance voltage
0x18810xA4	Cell11 balance voltage	Cell10 balance voltage	Cell9 balance voltage	Cell8 balance voltage
0x18810xA8	Cell7 balance voltage	Cell6 balance voltage	Cell5 balance voltage	Cell4 balance voltage
0x18810xAC	Cell3 balance voltage	Cell2 balance voltage	Cell1 balance voltage	Cell15 balance voltage
0x18810xB0	Cell16 balance voltage	Cell17 balance voltage	Cell18 balance voltage	N/A
0x18810x60	Primary temperature	Secondary temperature	Primary voltage reference	Secondary voltage reference
0x18810x64	Primary VAUX	Secondary VAUX	Primary VDDC	Secondary VDDC

BCC Status

CAN ID [Hex]	Byte[0:1]	Byte[2:3]	Byte[4:5]	Byte[6:7]
0x18840x00	Access error	System fault	Supply fault0	Supply fault0
0x18840x01	Analogy fault	Communication fault	Measurement fault	N/A
0x18840x02	Analogy OV fault0	Analogy OV0 fault0	Analogy UV1 fault0	Ainx OVUV fault
0x18840x03	Ialarm out reason	Wakeup reason	Power on reason	Analogy OV fault1
0x18840x04	Analogy UV0 fault1	Analogy UV1 fault	N/A	N/A

safety mechanism(SM) Information

CAN ID [Hex]	Byte[0:1]	Byte[2:3]	Byte[4:5]
0x18850000	SM 1-15	SM 16-30	SM 30-45

balance and physical status Information

CAN ID [Hex]	Byte[0:1]	Byte[2:3]	Byte[4:5]	Byte[6:7]
0x18870000	Logic balance	Physic balance	N/A	N/A

vcu data information

CAN ID [Hex]	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
0x18890000	gear	handBrake	pedal.accstate	pedal.accpos	pedal.brakeState	pedal.brakePos	torque	startKey

Figure 53. GUI interaction protocol

7 Revision history

Table 5. Revision history

Document ID	Release date	Description
UG10071 v.1.1	15 January 2024	Added IPCF
UG10071 v.1.0	5 June 2023	Initial release

8 Legal information

8.1 Definitions

Draft — A draft status on a document indicates that 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 in a draft version of a document and shall have no liability for the consequences of use of such information.

8.2 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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

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.

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

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. - NXP B.V. is not an operating company and it does not distribute or sell products.

8.3 Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

For more information, please visit: http://www.nxp.com

Date of release: 15 January 2024
Document identifier: UG10071