NXP Semiconductors

Document identifier: MPC5775BDCRDUG

Rev. 0, February 2020

MPC5775B BMS plus VCU Reference Design User Guide

Contents

Chapter 1 Introduction

This user guide gives an overview of hardware and software for the powertrain domain controller reference design based on MPC5775B MCU integrated the BMS and VCU in one ECU. The reference design is intended to provide a mechanism for easy customer evaluation of the Cobra55 family of microprocessors, and to facilitate hardware and software development.

1.1 List of acronyms

The following table provides a list and description of acronyms used throughout this document.

Table 1. Acronyms and abbreviations

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 Interface
PHY	Physical

Chapter 2 Hardware User Guide

2.1 Overview

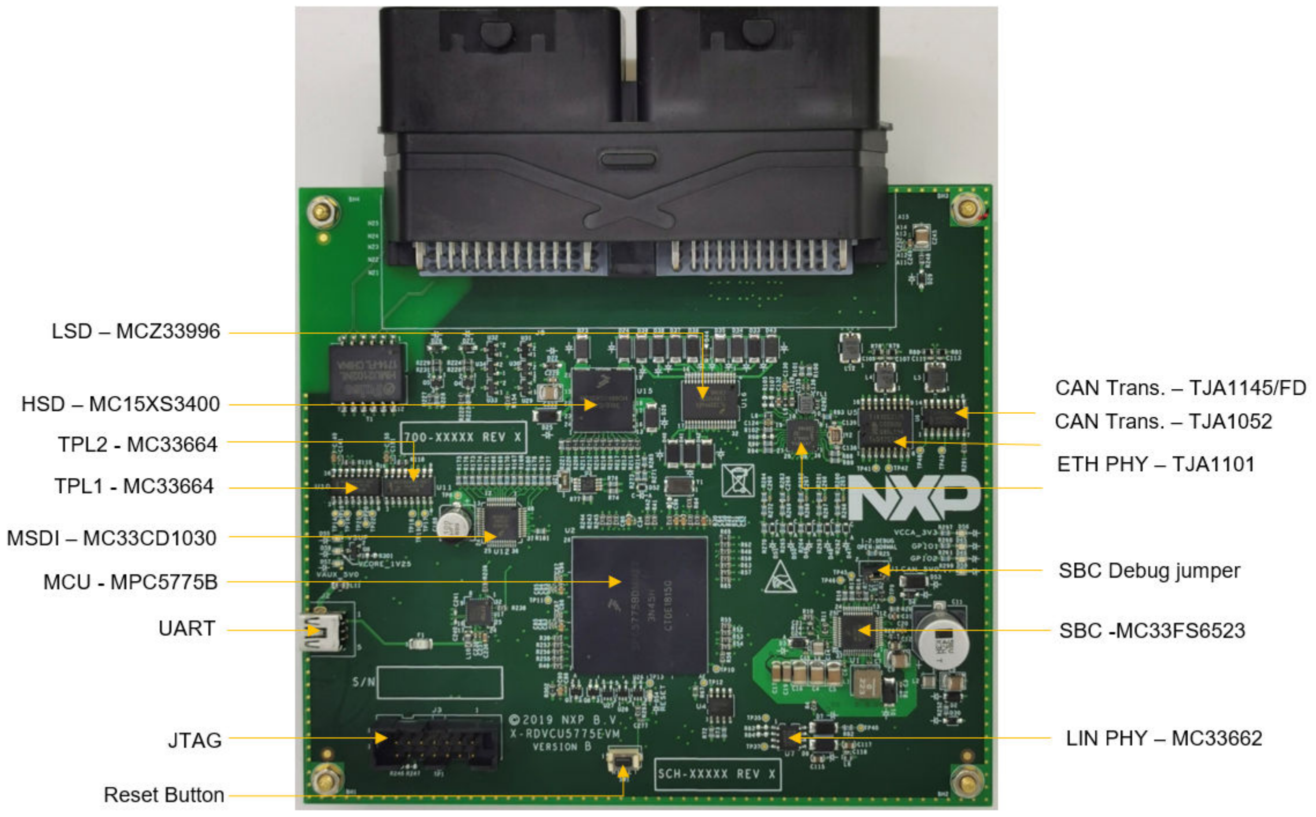


Figure 1. Master board

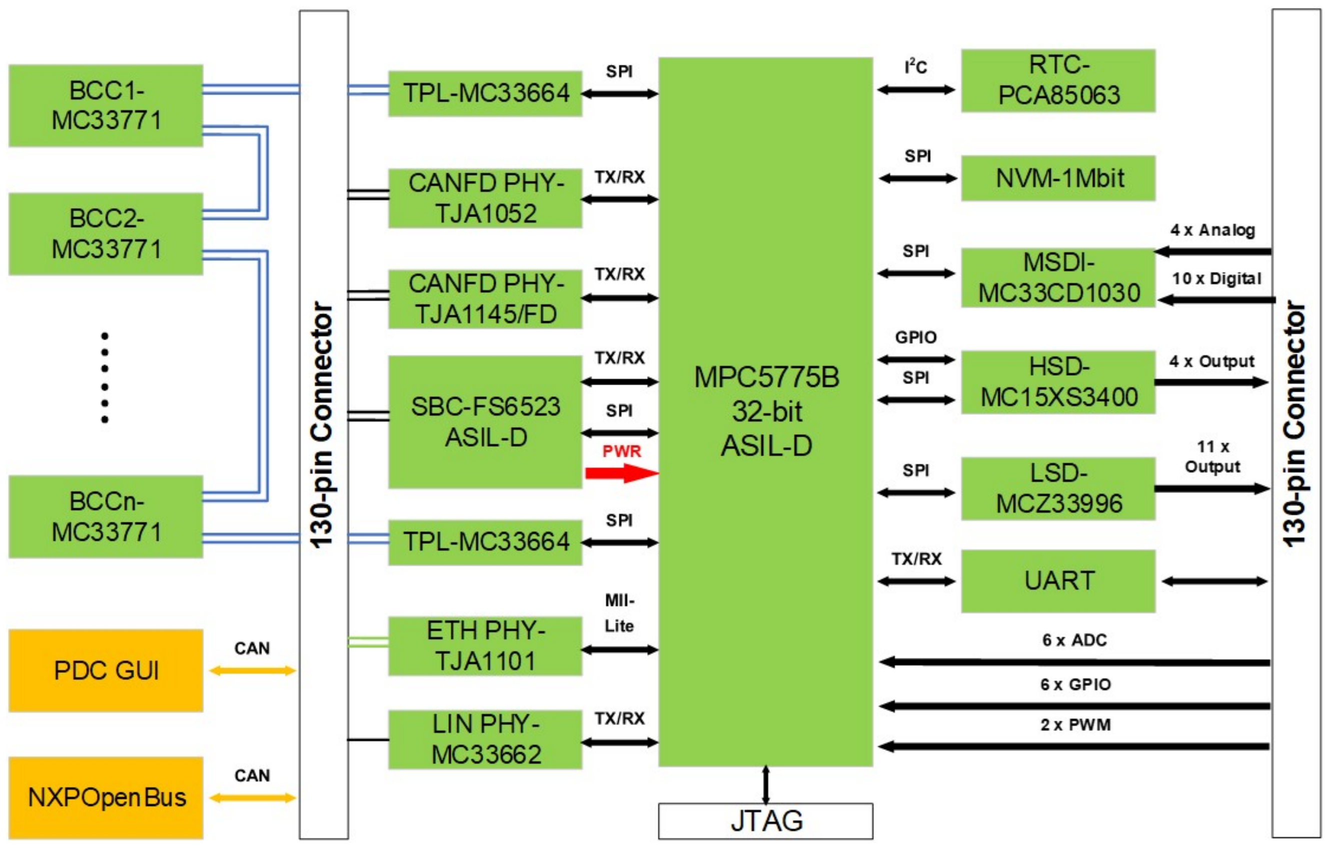


Figure 2. System block diagram

2.2 Board features

The system mainly consists of a master board and several slave boards connected in series with a daisy-chain with TPL.

The master board provides the following key features:

- NXP MPC5775B Microcontroller (416 MAPBGA soldered)
- Support Daisy chain of BCC device connection with Transformer isolation
- Integrate NXP Power SBC FS65xx with one CAN physical layer
- Master power regulator status LEDs
- One user reset switch with reset status LEDs
- Two user LEDs is connected to GPIO for test
- Standard 14-pin JTAG debug connector
- One Mbit Automotive external EEPROMs with SPI
- 1 Automotive tiny Real-Time Clock/calendar with alarm function and I²C-bus
- Integrate multiple communication interfaces:
- 1 x 100BASE-T1 Automotive Ethernet interface TJA1101
- 3 x CAN interfaces, two is supported via NXP CANFD transceiver TJA1052 and TJA1145, another is supported via FS65xx.
- 1 x LIN interface MC33662BLEF

- 1 x Mini USB / UART FDTI transceiver to interface with MCU
- Integrate input/output devices:
- 33 channel multiple switch detection interface devices(MC33CD1030)
- Quad high side with SPI Control(MC15XS3400)
- 16-output low side switch with SPI Control(MC33996)
- A 130-pin ECU connector, routing external I/O signals including:
- 10 x ADC input channels
- 16 x Switch input channels
- 2 x PWM input capture channels
- At least 2 x PWM output channels via HSD and LSD
- 5 x 5V + 1 x 12 V power supply for external sensor
- 4 x HSD output channels
- 11 x LSD output channels

NOTE

14 channels of MC33CD1030 is routed to 130-pin connector and 11 channels of MC33996 is routed to 130-pin connector.

2.3 Module introduction

2.3.1 Power SBC

2.3.1.1 Board power

The MC33FS6523CAE provides a robust, scalable power management to the MPC5775B MCU with fail silent safety monitoring measures making it fit for ASIL-D. It consists of multiple switching and linear voltage regulators.

When external power, typical 12 V automotive DC power supply with current limit > 600 mA, is applied to jumper J8 connector of master board.

- VBAT pins connects to DC power supplier 12 V+.
- GND pins connects to DC power supplier GND,
- Keep Jumper J1 onboard.

Five green power LEDs show the presence of the supply voltages as follows:

- LED D55 - Indicates that the 12 V external power is enabled and working correctly.
- LED D56 - Indicates that the 3.3 V linear regulator is enabled and working correctly.
- LED D57 - Indicates that the 5.0 V linear regulator is enabled and working correctly.
- LED D58 - Indicates that the 5.0 V linear regulator dedicated to the internal CAN FD interface is enabled and working correctly.
- LED D59 - Indicates that the 1.25 V switching regulator is enabled and working correctly.

The following figure shows the power LED's circuit in detail.

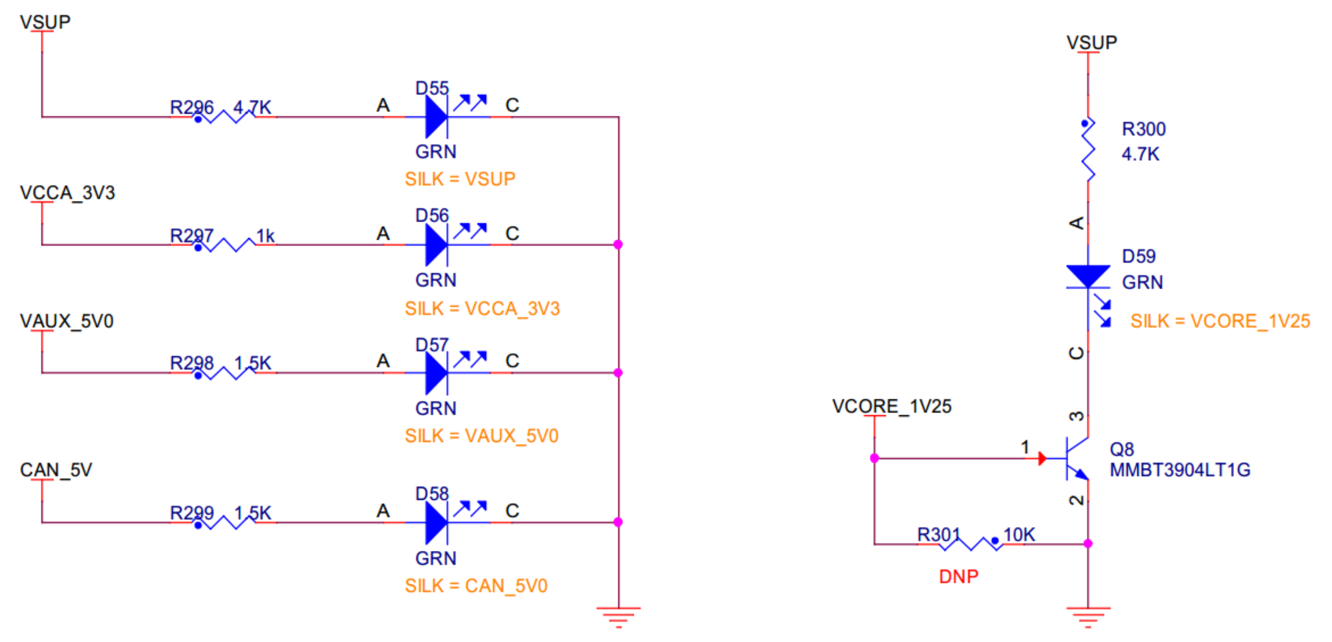


Figure 3. Board power LEDs

2.3.1.2 VCCA, VAUX voltage configuration

The voltage of V_{CCA} and V_{AUX} is configurable through the external resistor(**R8**) connected to **SELECT pin**. The default value of **R8** on the board is 24K, it means V_{CCA}=3.3 V, V_{AUX}=5.0 V. The following table show the configuration of V_{CCA}, V_{AUX} voltage, if the SELECT pin is detected open, the V_{CCA} regulators start at their minimum output voltage 3.3 V.

V_{CCA}/V_{AUX} voltage selection

Table 2. V_{CCA} and V_{AUX} configuration

V_{CCA}(V)	V_{AUX}(V)	R select	Recommended value
3.3	3.3	< 6.0 kΩ	5.1 kΩ ±5.0%
5.0	5.0	10.8 << 13.2 kΩ	12 kΩ ±5.0%
3.3	5.0	21.6 << 26.2 kΩ	24 kΩ ±5.0%
5.0	3.3	45.9 << 56.1 kΩ	51 kΩ ±5.0%

2.3.1.3 Debug mode

The jumper **J1** is used to configure the DEBUG mode of SBC, keep it onboard as shown in the following figure to activates the Debug mode. Once activated, there is no deep fail-safe state.

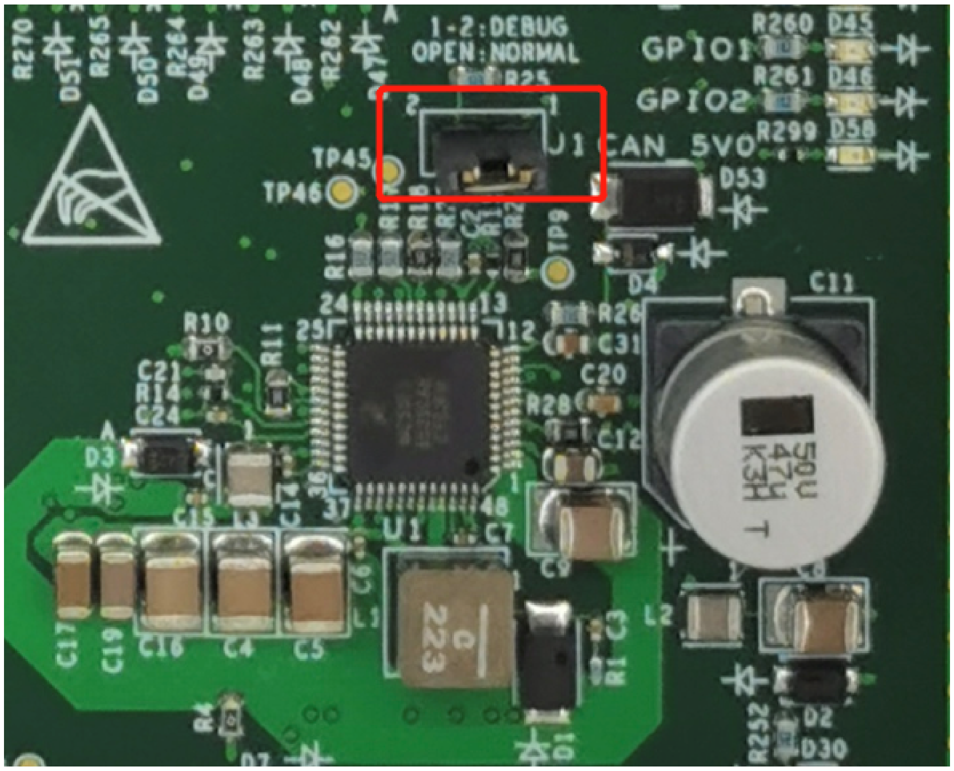


Figure 4. SBC DEBUG Jumper

2.3.1.4 Reset circuit

When the specific condition is reached, the SBC will reset MCU via SBC_RSTB pin. The following figure shows the board supports multiple MCU reset methods like SBC, JTAG and Switch Reset(SW1).

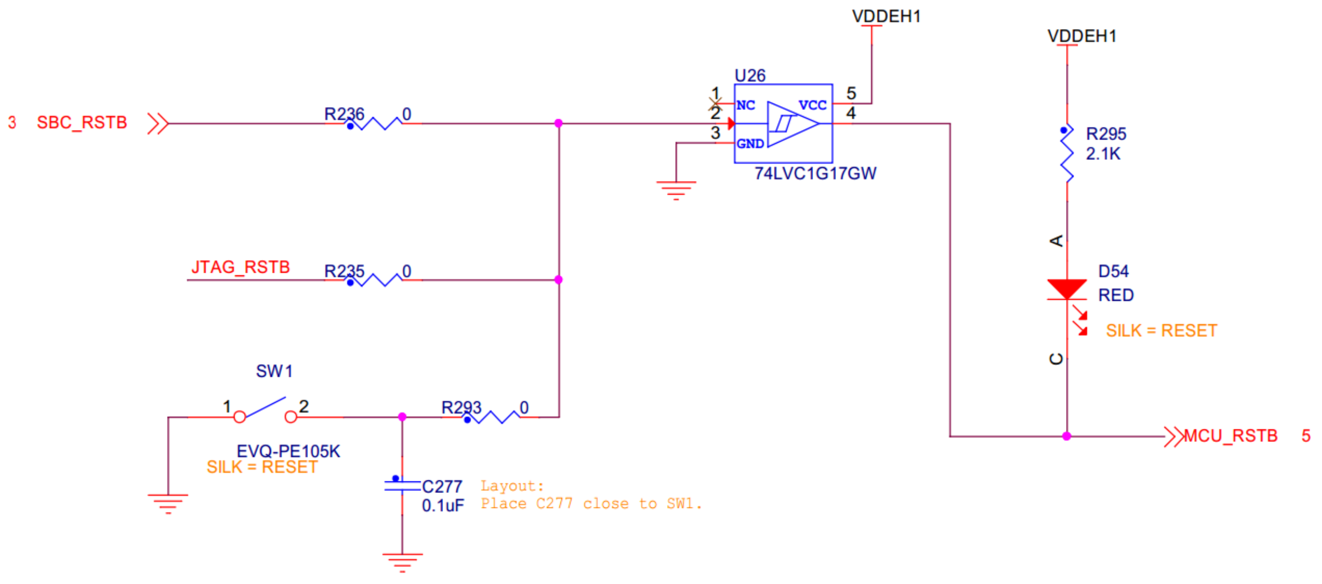


Figure 5. Board reset circuit

2.3.1.5 Communication interface

FS6523(U1) has a high speed CAN transceiver. It is routed to the MCU FlexCAN_B port(AC19, AD19). The pin-out of the 130-pin connector (J8) is C11, D11.

The SBC SPI signal is transferred to the MCU through DSPI2 with CS0. MCU uses the interface to control SBC watchdog and so on.

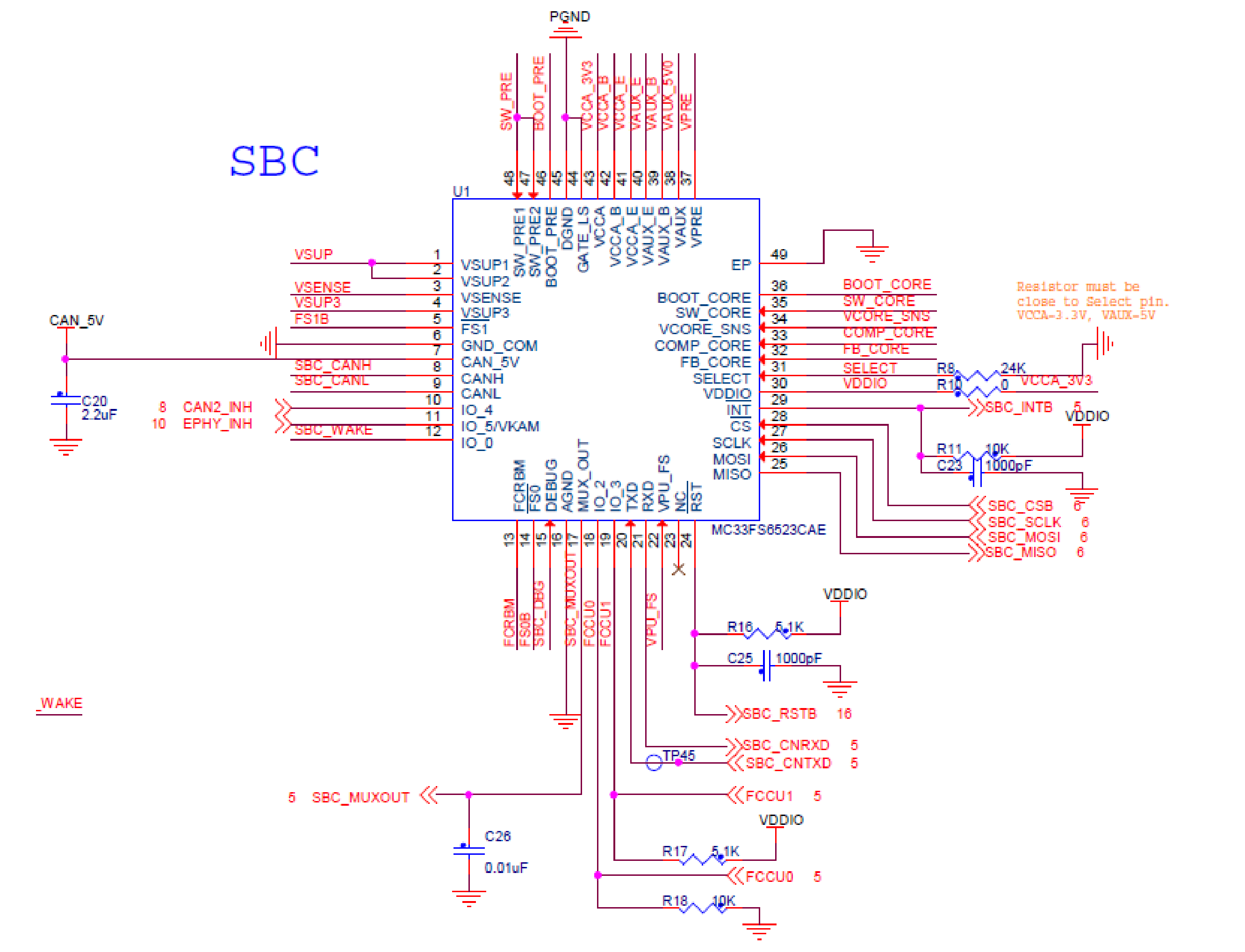


Figure 6. FS6523 application circuit

2.3.2 MCU power and GND configuration

The V_{DDEH} and V_{DDE} is MCU I/O supply voltage connected to V_{CCA}(3.3V) from SBC, and V_{DDA_EQ} is eQADC supply voltage connected to V_{AUX}(5V).

2.3.3 MCU external clock circuit

In addition to the internal 16 MHz oscillator, the MCU can also be clocked by external oscillator (Y1). The clock circuitry for the 40 MHz crystal is shown in the following figure. The symbol named “DNP” indicates the device is not mounted to the PCB.

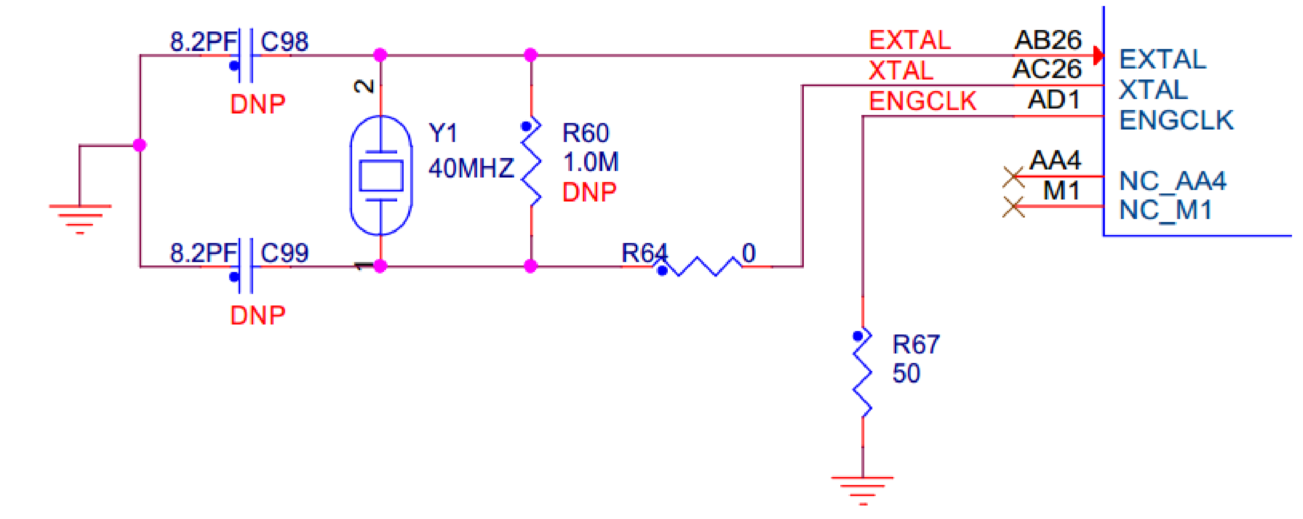


Figure 7. 40 MHz crystal circuit

2.3.4 JTAG connector

The master board is fitted with 14-pin JTAG debug connector. The following figure shows the 14-pin JTAG connector pinout.

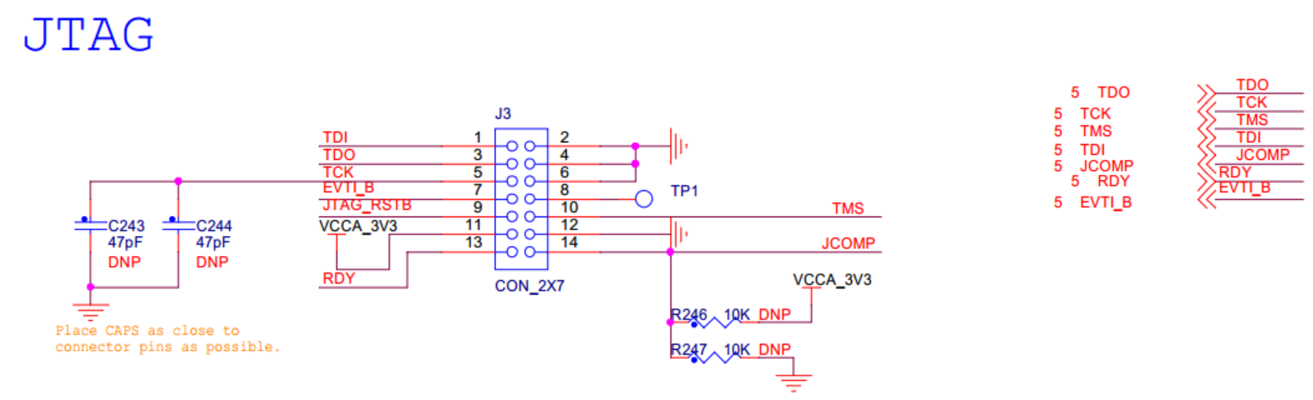


Figure 8. JTAG connector point

2.3.5 USB/UART transceiver

Mini USB connector interfaced with FT232RQ is connected to MCU eSCI_C port(AD22, AF23), USB to serial UART chip (U17). The USB to serial UART connection is shown in the following figure. Can connect to PC via mini USB to put string on screen or get string from screen by Serial Debug Assistant.

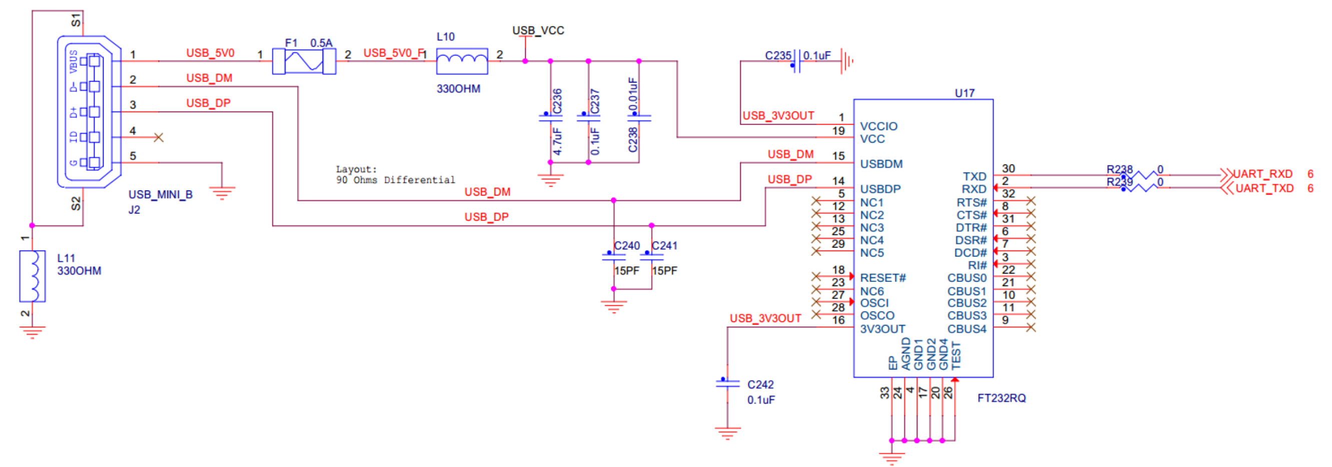


Figure 9. USB/UART transceiver circuit

2.3.6 CAN transceiver

The master board has two high speed CAN transceivers.

- TJA1052i(U5) is a galvanically isolated interface between CAN controller and the physical two-wire CAN bus, and connected to MCU MCAN0/FlexCAN_A port (AE19, AF19).
- TJA1045T/FD(U6) is a High-speed CAN transceiver for partial networking, and connected to MCU MCAN1/FlexCAN_D port (AC20, AD20). TJA1045T/FD must be drove through DSPI. The SPI signal is transferred to the MCU through DSPI0 with CS2, MCU use the interface to configure the PHY.

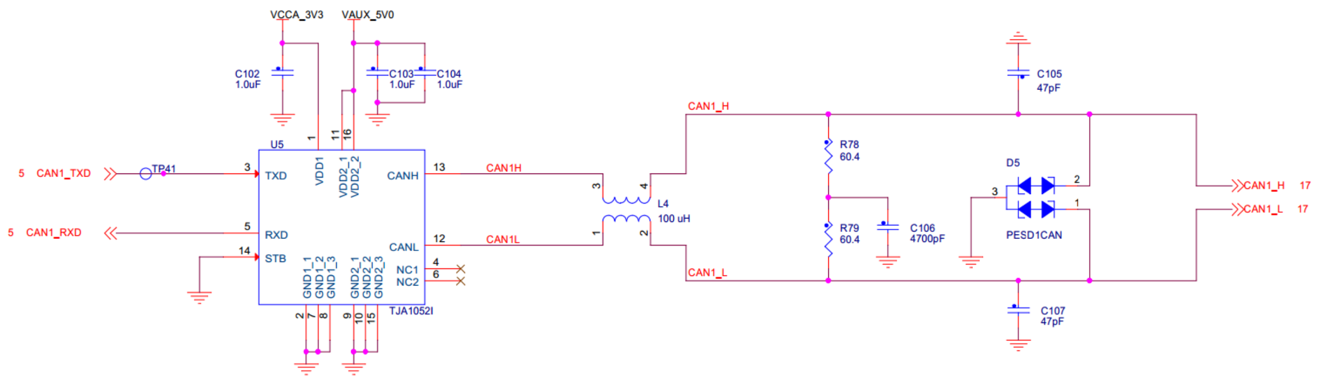


Figure 10. TJA1052i transceiver circuit

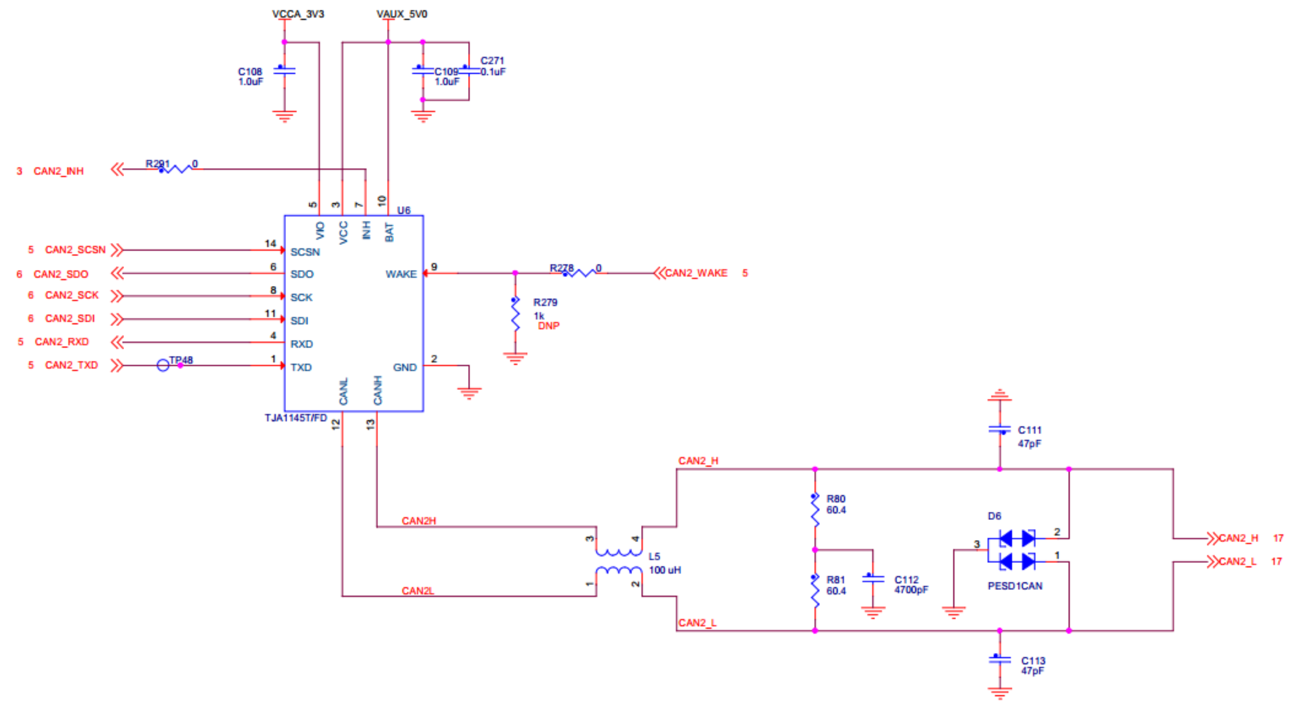


Figure 11. TJA1145 transceiver circuit

2.3.7 LIN transceiver

The master board is fitted with a NXP MC33662LEF LIN transceiver (U7) connected to MCU eSCI_A port(M2, M3). Configure MCU eSCI_A module and assert LIN1_EN pin to enable the LIN communication.

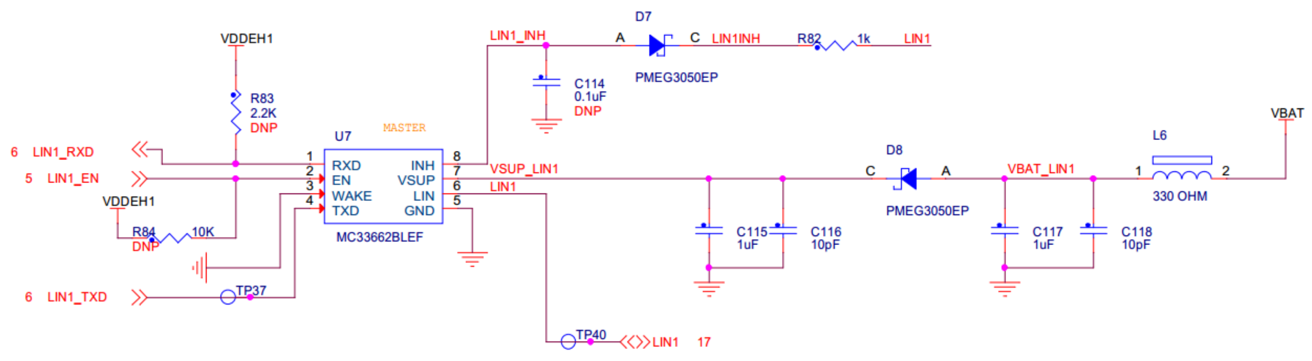


Figure 12. LIN transceiver circuit

2.3.8 Ethernet PHY

The master board includes support for 100M-base T1 Automotive Ethernet PHY TJA1101(U9), utilizing the normal MII lite. It is clocked by 25 MHz external oscillator (Y2). The default physical address is configured as 0b00xx0 by the pull-down or pull-up resistor on the pin.

A pair of differential signal(EPHY_TRX_P, EPHY_TRX_M) is routed to 130-pin ECU connector.

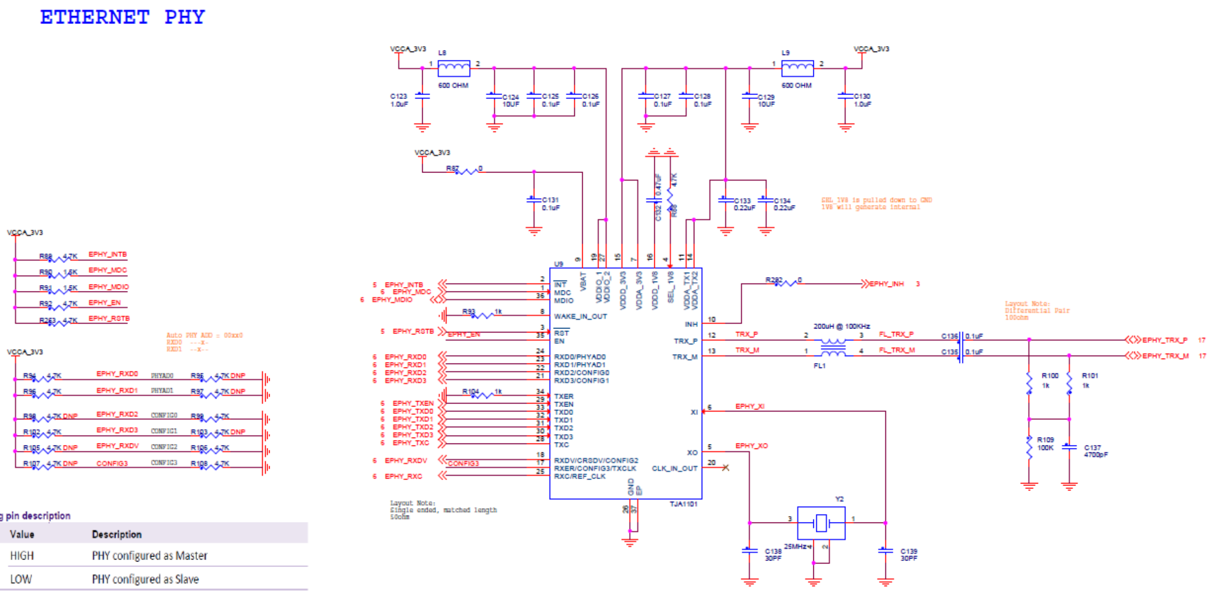


Table 9. Pin Strapping pin description

Pin	Value		Description
CONFIG0 (pin 22)	HIGH		PHY configured as Master
	LOW		PHY configured as Slave
CONFIG1 (pin 21)	HIGH		autonomous operation
	LOW		managed operation
CONFIG3/CONFIG2 (pin 17/pin 18)	LOW	LOW	Normal MII mode
	LOW	HIGH	RMII mode (external 50Mhz oscillator)
	HIGH	LOW	RMII mode (25MHz crystal)
	HIGH	HIGH	Reverse MII mode
SEL_1V8	HIGH		External 1.8V supply
	LOW		Internal 1.8V LDO enabled

Figure 13. TJA1101 ETH PHY circuit

2.3.9 eQADC interface

The master board enables nine eQADC channels from ANA0 to ANA8. The first three channels are routed to MSDI_AMUX, HSD_CSNS and SBC_MUXOUT port respectively. The others channels are routed to J8 to measure external analog signal, such as pedal, sensor signal, through ADC input signal process circuit (as shown in the following figure).

In addition to MCU_AIN4 channel supports external 0-12 V analog signal input from J8, other channels only support 0-5 V.

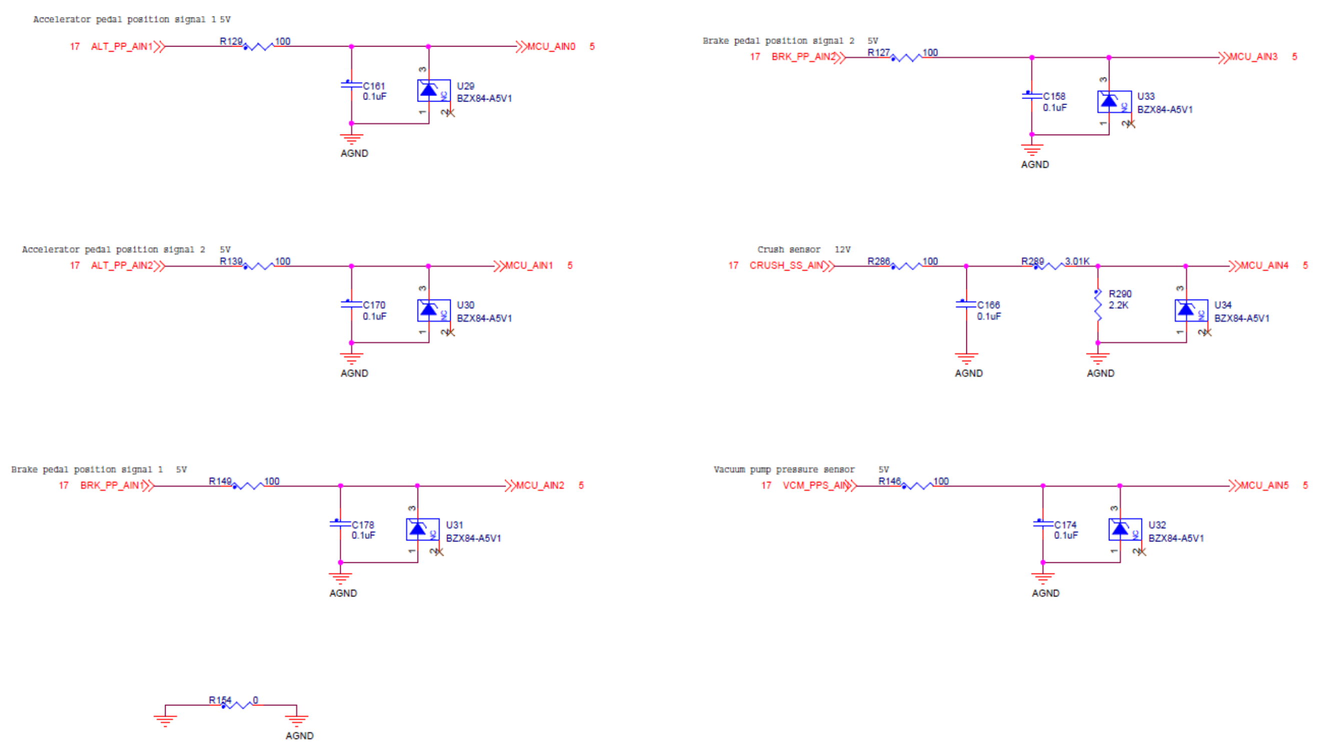


Figure 14. ADC input circuit

2.3.10 Digital signal interface

The master board supports to detect external switching signal, such as gear signal, through clamp circuit to MCU port (MCU_DIN0-MCU_DIN5) .

In addition to MCU_DIN4 and MCU_DIN5 channels support 0/12V digital signal input from J8, other channels only support 0/5V (as shown in the following figure).

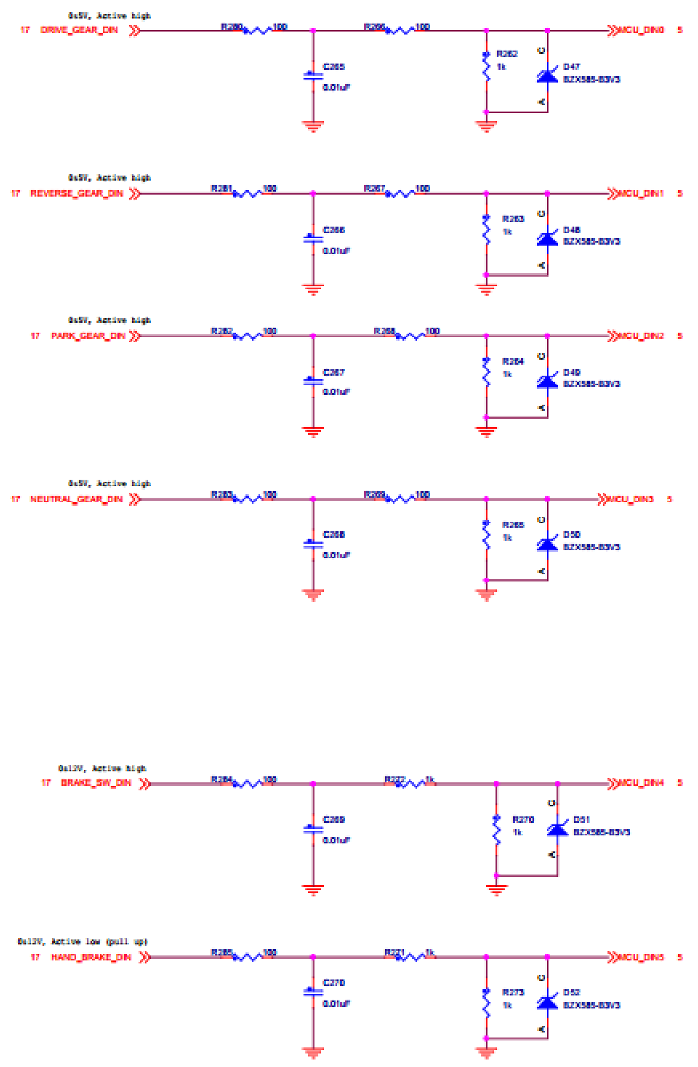


Figure 15. Digital input circuit

2.3.11 MSDI

The master board has a CD1030 to detect the closing and opening of multiple switch contacts and analog input signal. The input signal is transferred to the MCU through DSPI2 with CS1.

As shown in the following figure, the master board enables SG0-SG9 as digital input, SP0-SP4 as analog input, these port is connect to J8, and MCU read switch status via DPSI2, analog signal via MSDI_AMUX through eQADC channel 0 (ANA0).

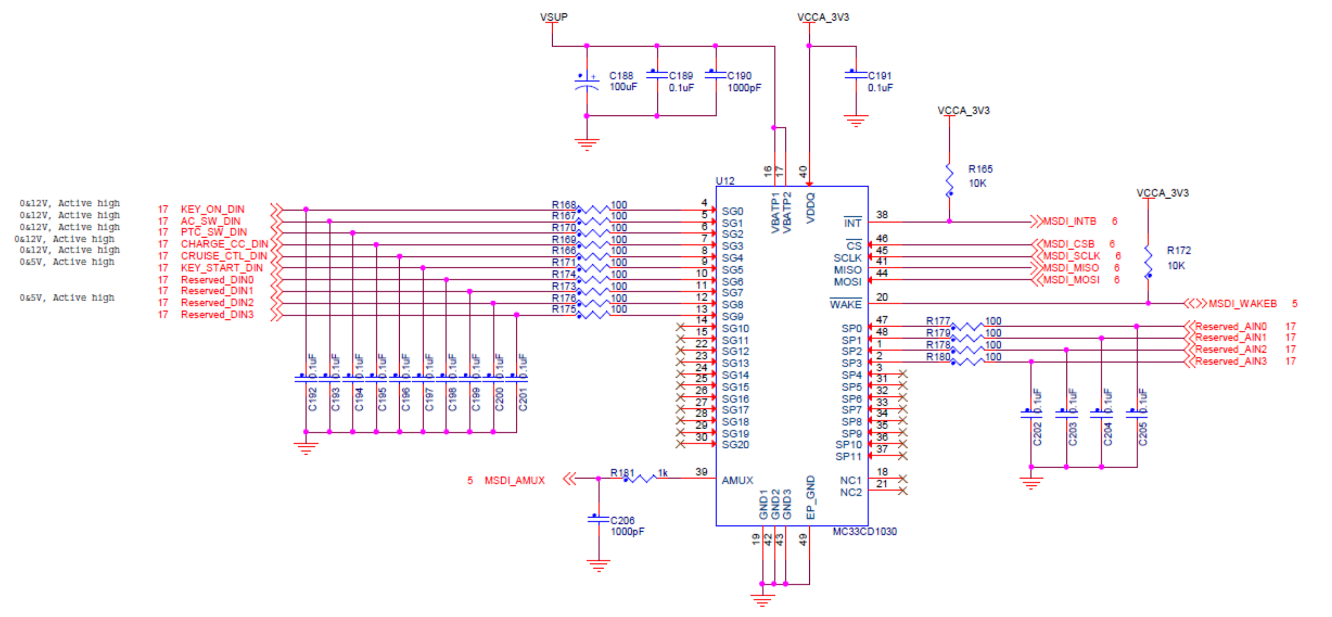


Figure 16. MSDI circuit

2.3.12 PWM input

The master board has two PWM input circuit to detect external PWM signal, the MCU_PWM_IN0/1 is routed to MCU eMIOS channels 1/2. The user can use eMIOS mode Pulse width measurement mode to measure PWM signal, its amplitudes range from 3.3 V to 12 V.

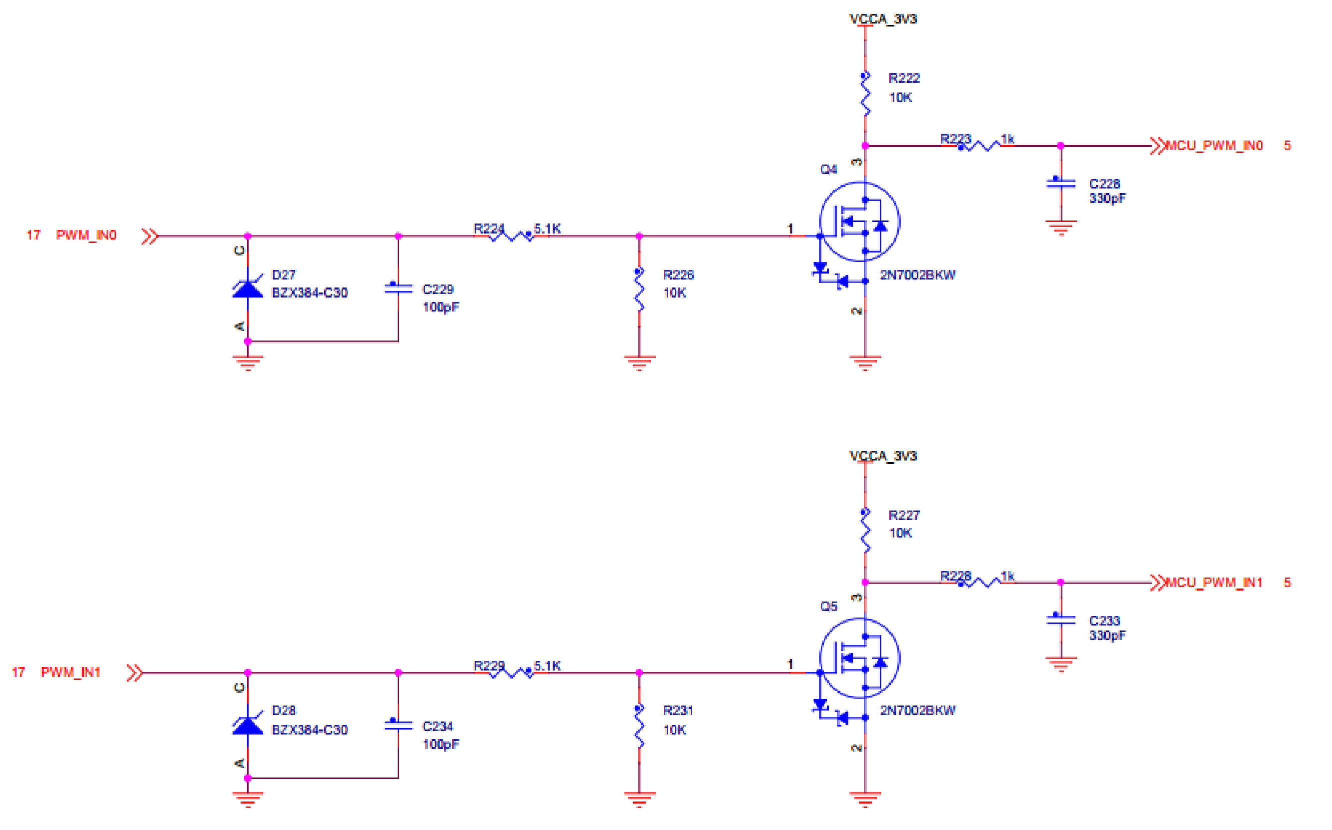


Figure 17. PWM circuit

2.3.13 HSD (extreme switch)

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

Generally load as:

- Main Contactor - 12 V dc coil voltage rating, 1~2A coil current at ON.
- Pre-Charge Contactor - 12 V dc coil voltage rating, 0.5 A coil current at ON.
- Cooling Fans - 1 2 V dc, 4.8 A
- Heater - 12 V dc 40 W

The master board use an extreme switch devices, NXP Quad high-side switch MC15XS3400(U15), to drive these loads, it is supplied by 12 V dc power. It can provide four channel high side outputs, each channel up to 6 A current.

MCU can configure DSPI2 with CS2 to control it, and also supports the direct control to the output pins (HS0-HS3) via the HSD_IN0-HSD_IN3 with GPIO.

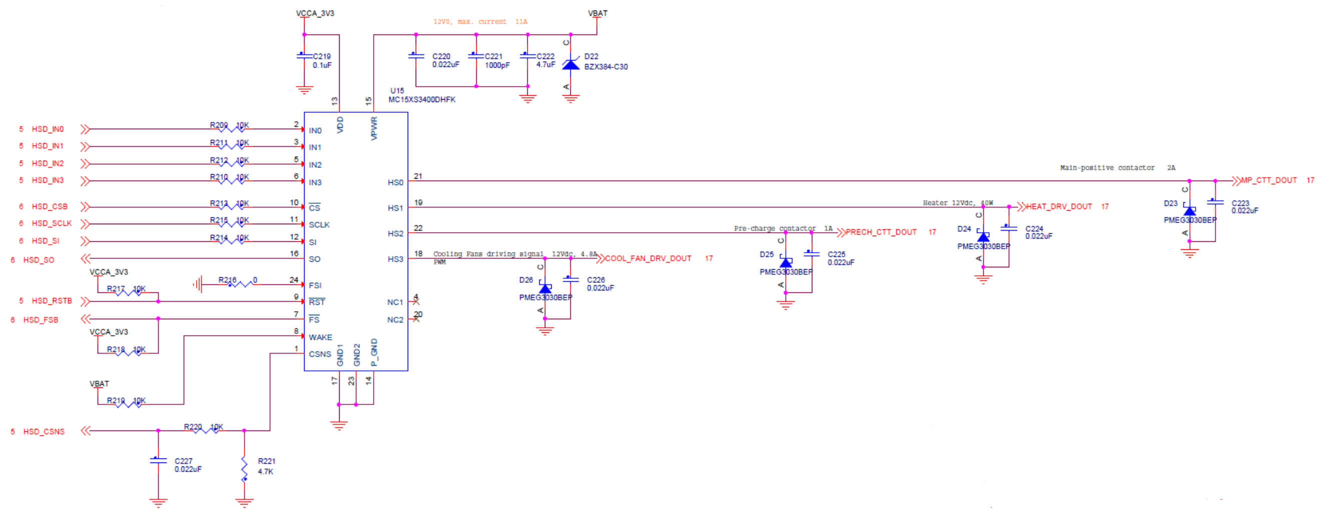


Figure 18. HSD circuit

2.3.14 LSD

The system requires several channels to control external contactor or relays, such as AC power switch, DCDC relay, etc. So the master board is fitted with a NXP 16-output low side switch with a 24-bit serial input control MC33996 (U16), MCU can configure DSPI2 with CS3 to control it. The PWM pin – LSD_PWM is routed MCU eMIOS channel three and the output of the output PWM frequency is up to 2.0 KHz.

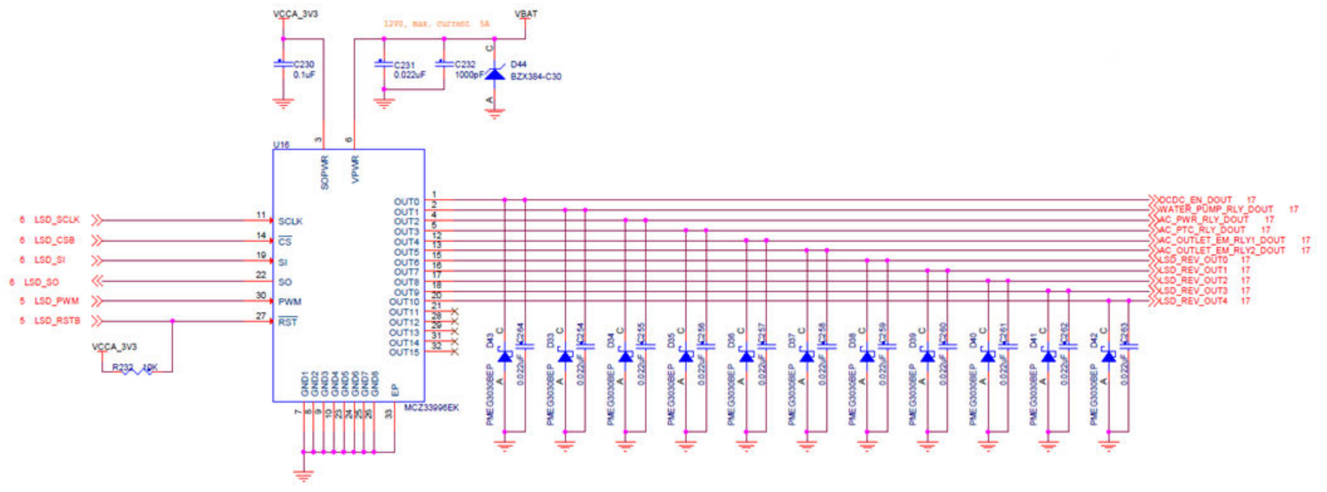


Figure 19. LSD circuit

2.3.15 TPL

Due to high voltage of battery packs, isolation is very important. This reference design isolate power supply and communications. TPL communication daisy chain between slave boards is above 3750 V dc by device.

The MC33664 is a transceiver physical layer transformer driver designed to interface a MCU conveniently to a high speed isolated communication network. It is needed for dual SPI for communication with MCU by half-duplex.

TPL communication port between master board and slave boards, the baud rate is 2 Mbps. It is linked as daisy chain with one or more slave board. The board has two TPL(U10, U11) routed three MCU DSPI, includes DSPI0 with CS0 for master, DSPI1 with CS0 for TPL1 slave, DSPI3 with CS0 for TPL2 slave.

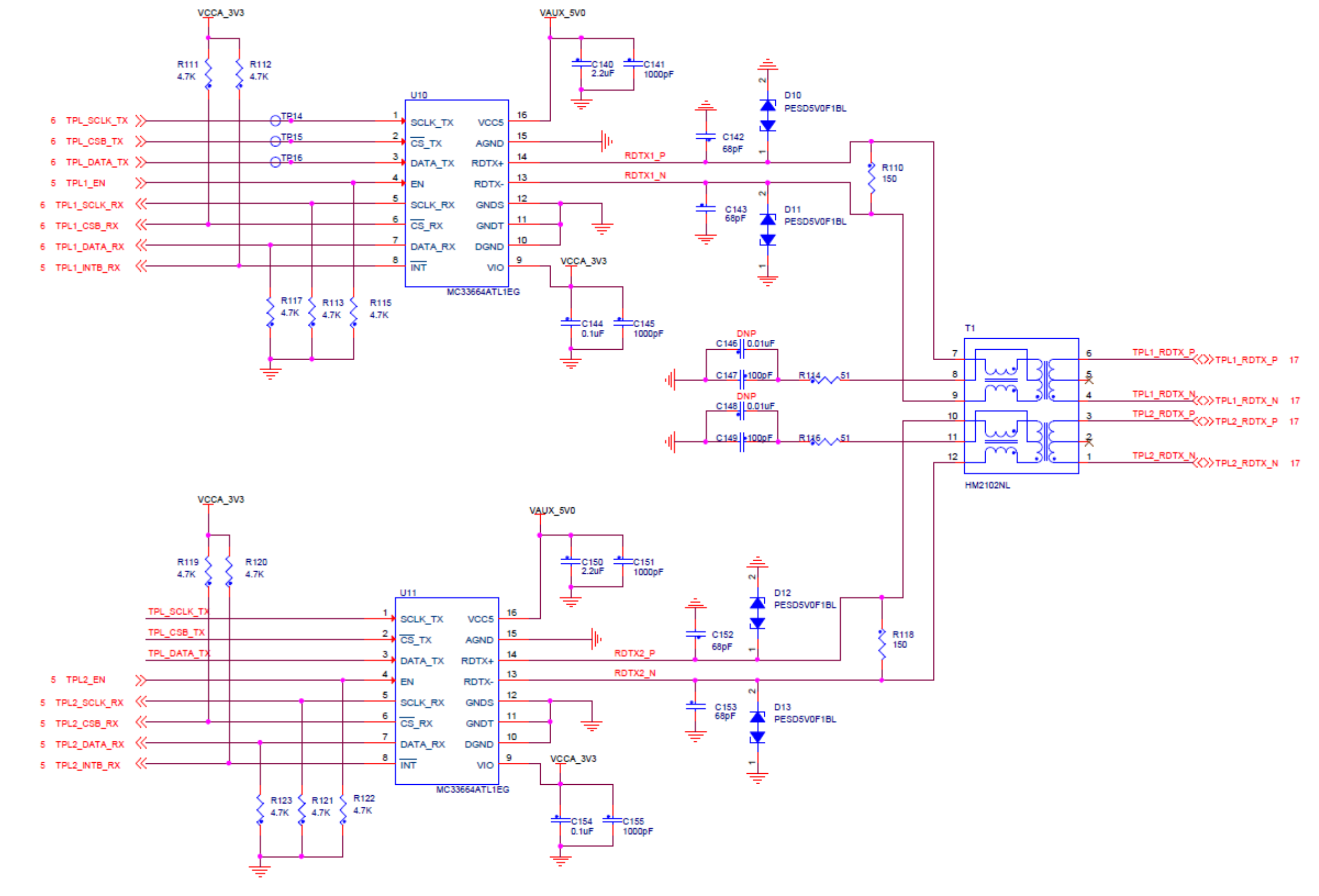


Figure 20. TPL circuit

2.3.16 RTC

The master board is fitted with a NXP automotive RTC PCA85063(U3) with I^{2}C communication, but the MPC5775B has no I^{2}C interface, so it can only be implemented by using the GPIO(V25, V26) to simulate I^{2}C protocol. The default devices address is 0x51.

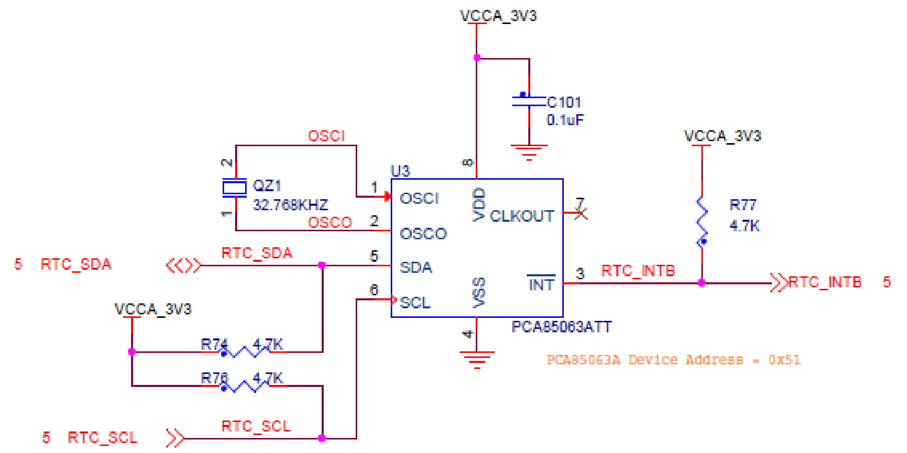


Figure 21. RTC circuit

2.3.17 NVM

The master board has an external NVM – M95M01, an automotive 1024 Kbit SPI bus EEPROM with high speed clock. MCU can read and write data through DSPI0 with CS1.

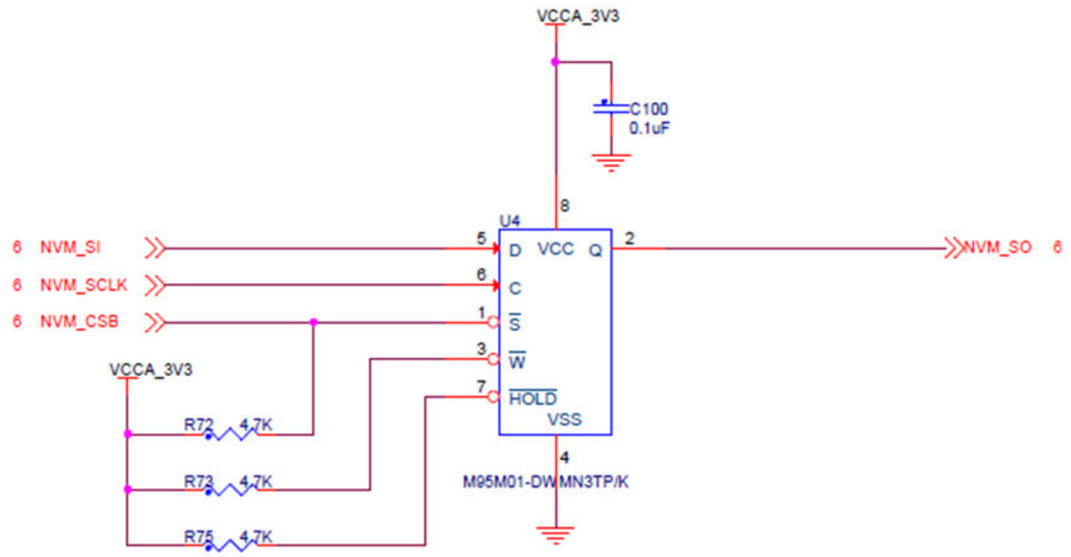


Figure 22. NVM circuit

2.4 130 pin ECU connector

There is a 130-pin ECU connector onboard, consisting of two parts of header M1 and M2.

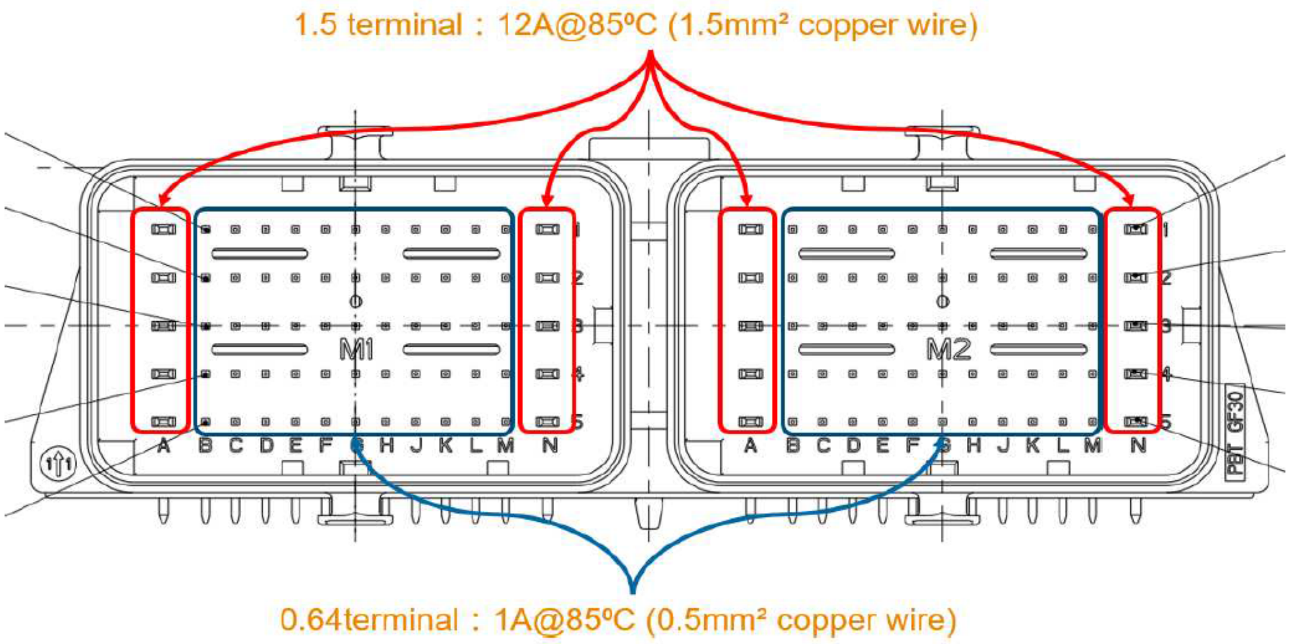

1.5 terminal：12A@85ºC (1.5mm² copper wire)
M1
M2
A B C D E F G H J K L M N
1 2 3 4 5
PBT GF30
0.64terminal：1A@85ºC (0.5mm² copper wire)

Figure 23. 130-pin ECU connector

When the connector is inserted into the cable, the two supporting cables are installed in sequence. The M2 corresponding cable has two pairs of differential twisted pair wires for connecting the BCC slave board. The following figure show the cables installation.

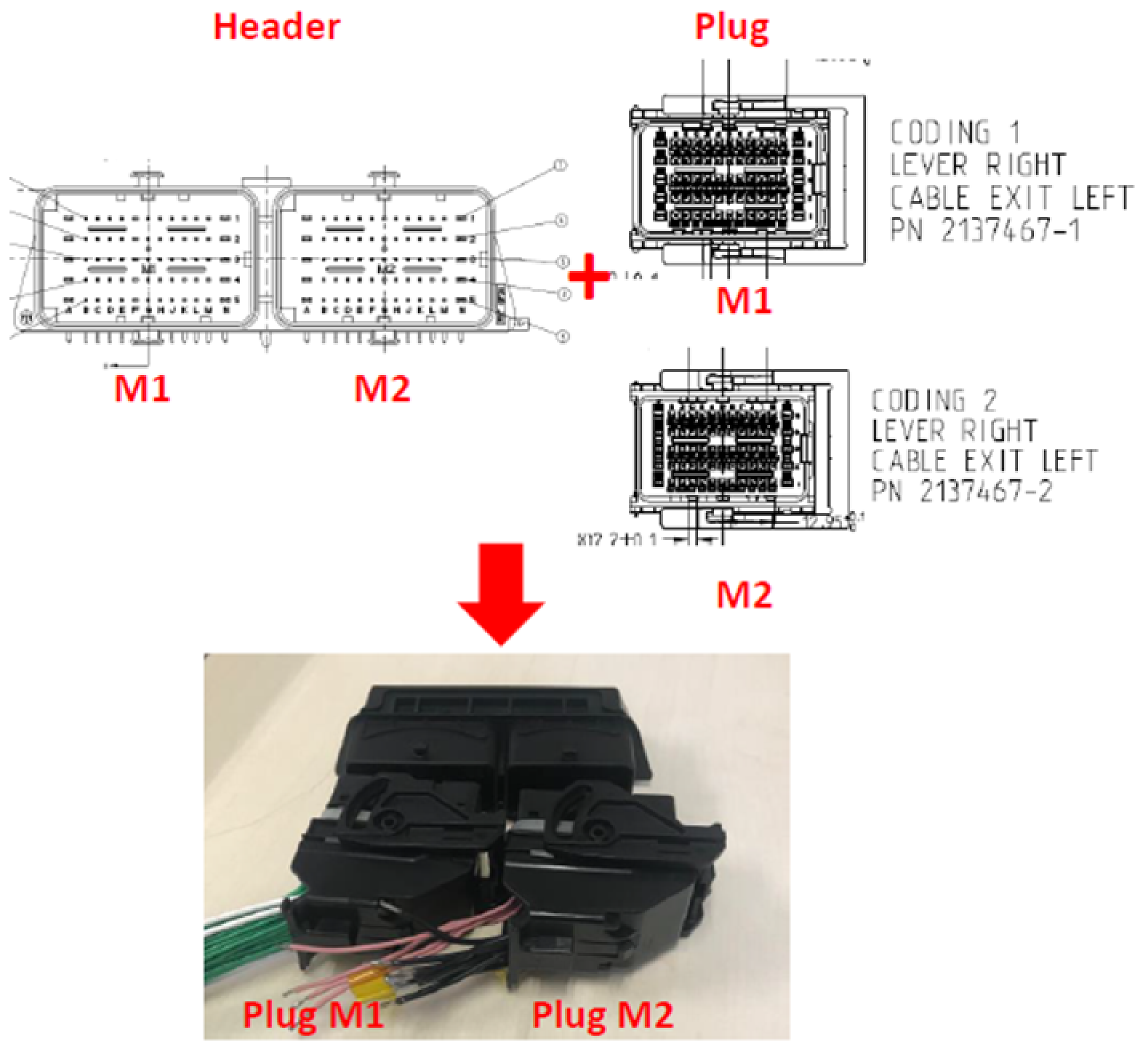


Figure 24. 130-pin ECU Cable Installation diagram

The following table lists all the connections to the master board interface connector .

Table 3. Board interface connector details

Connector	Signal	Description	Connector	Signal	Description
A11	VBAT	Power supply	A21	GND	Digital GND
A12	VBAT	Power supply	A22	GND	Digital GND
A13	VBAT	Power supply	A23	GND	Digital GND
A14	GND	Digital GND	A24	GND	Digital GND
A15	GND	Digital GND	A25	GND	Digital GND

Table continues on the next page...

Table 3. Board interface connector details (continued)

Connector	Signal	Description	Connector	Signal	Description
B11	GND	Digital GND	B21	AGND	Analog GND
B12	GND	Digital GND	B22	AGND	Analog GND
B13	GND	Digital GND	B23	AGND	Analog GND
B14	GND	Digital GND	B24	AGND	Analog GND
B15	GND	Digital GND	B25	AGND	Analog GND
C11	SBC_CAN_L	Flex CAN1	C21	Reserved_AIN0	MSDI SP0
C12	CAN1_L	MCAN0	C22	Reserved_AIN1	MSDI SP1
C13	CAN2_L	MCAN1	C23	CRUSH_SS_AIN	Crush sensor signal
C14	NC	Not connected	C24	Reserved_AIN2	MSDI SP2
C15	LIN1	LIN signal output	C25	Reserved_AIN3	MSDI SP3
D11	SBC_CAN_H	Flex CAN1	D21	ALT_PP_AIN1	Accelerator pedal signal 1
D12	CAN1_H	MCAN0	D22	ALT_PP_AIN2	Accelerator pedal signal 2
D13	CAN2_H	MCAN1	D23	BRK_PP_AIN1	Brake pedal signal 1
D14	NC	Not connected	D24	BRK_PP_AIN2	Brake pedal signal 2
D15	KL15_WAKE	SBC IO0 (Wake-UP pin)	D25	VCM_PPS_AIN	Vacuum pump pressure sensor
E11	GND	Digital GND	E21	AGND	Analog GND
E12	DRIVE_GEAR_DIN	Driver gear switch input	E22	AGND	Analog GND
E13	REVERSE_GEAR_DIN	Reverse gear switch input	E23	AGND	Analog GND
E14	PARK_GEAR_DIN	Parking gear switch input	E24	AGND	Analog GND
E15	NEUTRAL_GEAR_DIN	Neutral gear switch input	E25	AGND	Analog GND
F11	EPHY_TRX_P	Ethernet Differential signal (+)	F21	GND	Digital GND
F12	GND	Digital GND	F22	GND	Digital GND

Table continues on the next page...

Table 3. Board interface connector details (continued)

Connector	Signal	Description	Connector	Signal	Description
F13	GND	Digital GND	F23	GND	Digital GND
F14	BRAKE_SW_DIN	brake switch input	F24	GND	Digital GND
F15	HAND_BRAKE_DIN	Hand brake switch input	F25	GND	Digital GND
G11	EPHY_TRX_M	Ethernet Differential signal (-)	G21	KEY_ON_DIN	MSDI SG0
G12	GND	Digital GND	G22	AC_SW_DIN	MSDI SG1
G13	ALT_PS_5V0	5V supply for Accelerator pedal	G23	CRUISE_CTL_DIN	MSDI SG4
G14	ALT_PS_5V0	5V supply for Accelerator pedal	G24	CHARGE_CC_DIN	MSDI SG3
G15	GND	Digital GND	G25	PTC_SW_DIN	MSDI SG2
H11	GND	Digital GND	H21	PWM_IN0	PWM input
H12	VCM_PPS_5V0	5V supply for vacuum pump pressure sensor	H22	PWM_IN1	PWM input
H13	BRK_PS_5V0	5V supply for Brake pedal	H23	GND	Digital GND
H14	BRK_PS_5V0	5V supply for Brake pedal	H24	GND	Digital GND
H15	CRUSH_SS_12V	12V supply for crush sensor	H25	GND	Digital GND
J11	GND	Digital GND	J21	KEY_START_DIN	MSDI SG5
J12	GND	Digital GND	J22	Reserved_DIN0	MSDI SG6
J13	GND	Digital GND	J23	Reserved_DIN1	MSDI SG7
J14	GND	Digital GND	J24	Reserved_DIN2	MSDI SG8
J15	GND	Digital GND	J25	Reserved_DIN3	MSDI SG9
K11	AC_PWR_RLY_DOUT	LSD output 2	K21	NC	Not connected
K12	WATER_PUMP_RLY_DOUT	LSD output 1	K22	NC	Not connected
K13	DCDC_EN_DOUT	LSD output 0	K23	NC	Not connected

Table continues on the next page...

Table 3. Board interface connector details (continued)

Connector	Signal	Description	Connector	Signal	Description
K14	LSD_REV_OUT4	LSD output 10	K24	NC	Not connected
K15	LSD_REV_OUT3	LSD output 9	K25	NC	Not connected
L11	AC_OUTLET_EM_RLY2_DOUT	LSD output 5	L21	NC	Not connected
L12	LSD_REV_OUT2	LSD output 8	L22	NC	Not connected
L13	AC_OUTLET_EM_RLY1_DOUT	LSD output 4	L23	NC	Not connected
L14	LSD_REV_OUT0	LSD output 6	L24	NC	Not connected
L15	LSD_REV_OUT1	LSD output 7	L25	NC	Not connected
M11	GND	Digital GND	M21	TPL2_RDTX_N	TPL2 Differential signal (-)
M12	GND	Digital GND	M22	TPL2_RDTX_P	TPL2 Differential signal (+)
M13	GND	Digital GND	M23	NC	Not connected
M14	GND	Digital GND	M24	TPL1_RDTX_N	TPL1 Differential signal (-)
M15	GND	Digital GND	M25	TPL1_RDTX_P	TPL1 Differential signal (+)
N11	AC_PTC_RLY_DOUT	LSD output 3	N21	NC	Not connected
N12	COOL_FAN_DRV_DOUT	HSD output 3	N22	NC	Not connected
N13	HEAT_DRV_DOUT	HSD output 1	N23	NC	Not connected
N14	PRECH_CTT_DOUT	HSD output 2	N24	NC	Not connected
N15	MP_CTT_DOUT	HSD output 0	N25	NC	Not connected

Chapter 3 Software User Guide

This reference design provides two sample application projects.

- S32 SDK(support FreeRTOS)
- AUTOSAR MCAL

3.1 SDK project

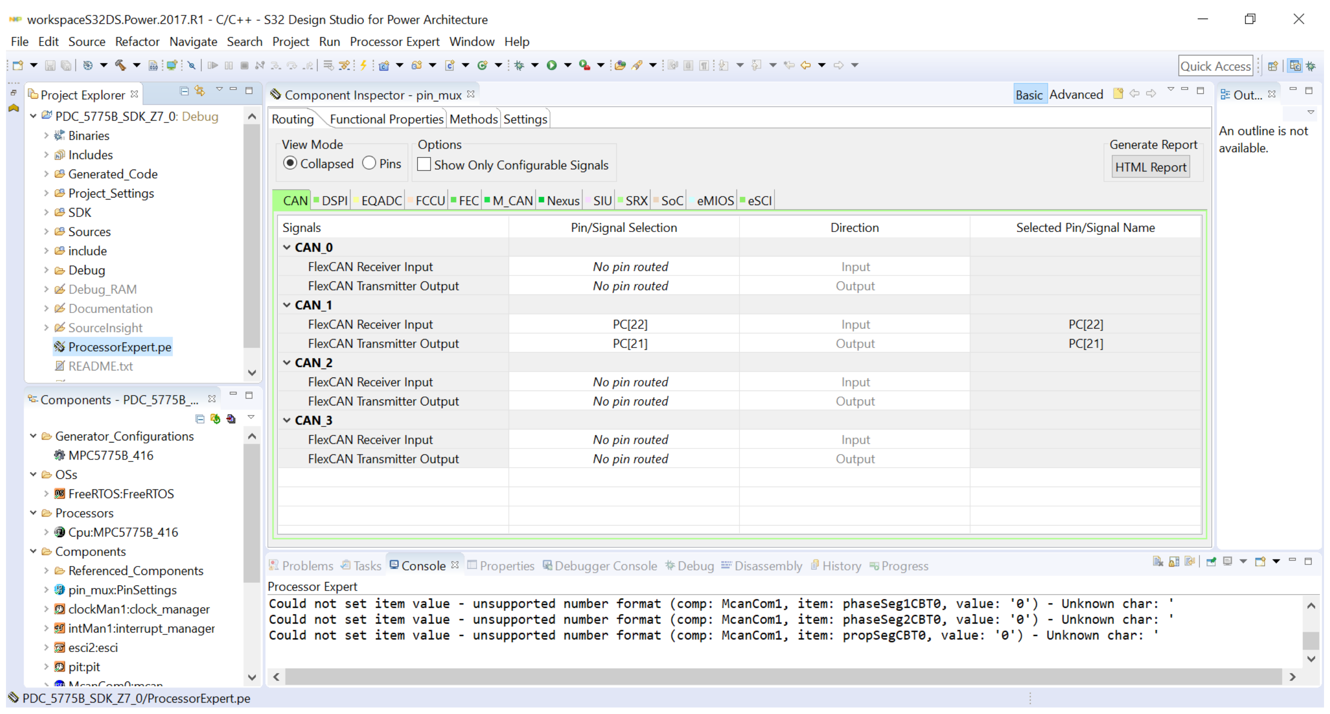
3.1.1 Software and tools requirement

The following is list of tools and software required for SDK project.

- IDE: S32 Design Studio for Power$^{®}$ Architecture Version 2017.R1.
- SDK: S32_SDK_S32PA_RTM_3.0.0.
- Compiler: GCC E200 VLE GNU Compiler 4.9.4.
- Debugger: P&E multilink (with P&E GDB Server) , Lauterbach TRACE32 JTAG debugger (use *run.cmm* script to debug).
- CAN Adaptor: PCAN-USB Pro.

NOTE

There is an issue in SDK drivers source code. Please modify the macro named PIT_CLOCK_NAMES from PITRTI0_CLK to PER_CLK in line 178 of the *MPC5775B_features.h* header file in which SDK/platform/devices/MPC5775B/include/.


workspaceS32DS.Power.2017.R1 - C/C++ - S32 Design Studio for Power Architecture
File Edit Source Refactor Navigate Search Project Run Processor Expert Window Help
Quick Access
Project Explorer
PDC_5775B_SDK_Z7_0: Debug
Binaries
Includes
Generated_Code
Project_Settings
SDK
Sources
include
Debug
Debug_RAM
Documentation
SourceInsight
ProcessorExpert.pe
README.txt
Components - PDC_5775B_...
Generator_Configurations
MPC5775B_416
OSs
FreeRTOS:FreeRTOS
Processors
Cpu:MPC5775B_416
Components
Referenced_Components
pin_mux:PinSettings
clockMan1:clock_manager
intMan1:interrupt_manager
esci2:esci
pit:pit
Component Inspector - pin_mux
Basic Advanced
Routing Functional Properties Methods Settings
View Mode
Options
Generate Report
HTML Report
Out...
An outline is not available.
CAN DSPI EQADC FCCU FEC M_CAN Nexus SIU SRX SoC eMIOS eSCI
Problems Tasks Console Properties Debugger Console Debug Disassembly History Progress
PDC_5775B_SDK_Z7_0/ProcessorExpert.pe

View Mode: [✓ Collapsed] Pins

Options: Show Only Configurable Signals

Signals	Pin/Signal Selection	Direction	Selected Pin/Signal Name
CAN_0			
FlexCAN Receiver Input	*No pin routed*	Input	
FlexCAN Transmitter Output	*No pin routed*	Output	
CAN_1			
FlexCAN Receiver Input	PC[22]	Input	PC[22]
FlexCAN Transmitter Output	PC[21]	Output	PC[21]
CAN_2			
FlexCAN Receiver Input	*No pin routed*	Input	
FlexCAN Transmitter Output	*No pin routed*	Output	
CAN_3			
FlexCAN Receiver Input	*No pin routed*	Input	
FlexCAN Transmitter Output	*No pin routed*	Output	

Processor Expert

```
Could not set item value - unsupported number format (comp: McanCom1, item: phaseSeg1CBT0, value: '0') - Unknown char: '
Could not set item value - unsupported number format (comp: McanCom1, item: phaseSeg2CBT0, value: '0') - Unknown char: '
Could not set item value - unsupported number format (comp: McanCom1, item: propSegCBT0, value: '0') - Unknown char: '
```

Figure 25. S32DS IDE

Figure 26. P&E multilink

Figure 27. PCAN-USB pro

3.1.2 File structure

There are two folders in the top-level directory named PDC_5775B_SDK, which contain two projects - core 0 projects (in PDC_5775B_SDK_Z7_0 folder) and core 1 projects (in PDC_5775B_SDK_Z7_1 folder). The provided demo codes only run on core 0. The following table shows the file structure.

Table 4. SDK project files

Folders/Files	Descriptions
.settings	contains customized project settings.
bin	contains generated object file.
bin/bootloader_bin	contains bootloader binary files.
Documentation	contains automatically generated documents by S32DS.

Table continues on the next page...

Table 4. SDK project files (continued)

Folders/Files	Descriptions
Generated_Code	contains all generated codes by ProcessorExpert.pe tool.
include	contains the application header files with pdc_ prefix.
Project_Settings	contains two version (bootloader and non-bootloader) files - linker, startup code and debugger configurations.
SDK	contains the SDK source codes used in this project.
Sources	contains the application source codes, related drivers folders.
ProcessorExpert.pe file	ProcessorExpert.pe tool.
run.cmm	is a script used for Lauterbach debugger

NOTE

Please unpack "Bootloader_Settings.zip" in Project_Settings folder to replace *linker_flash.ld* and *startup_MPC5775B.S* for bootloader support project. If you do not want to enable bootloader, unpack "Non_Bootloader_Settings.zip".

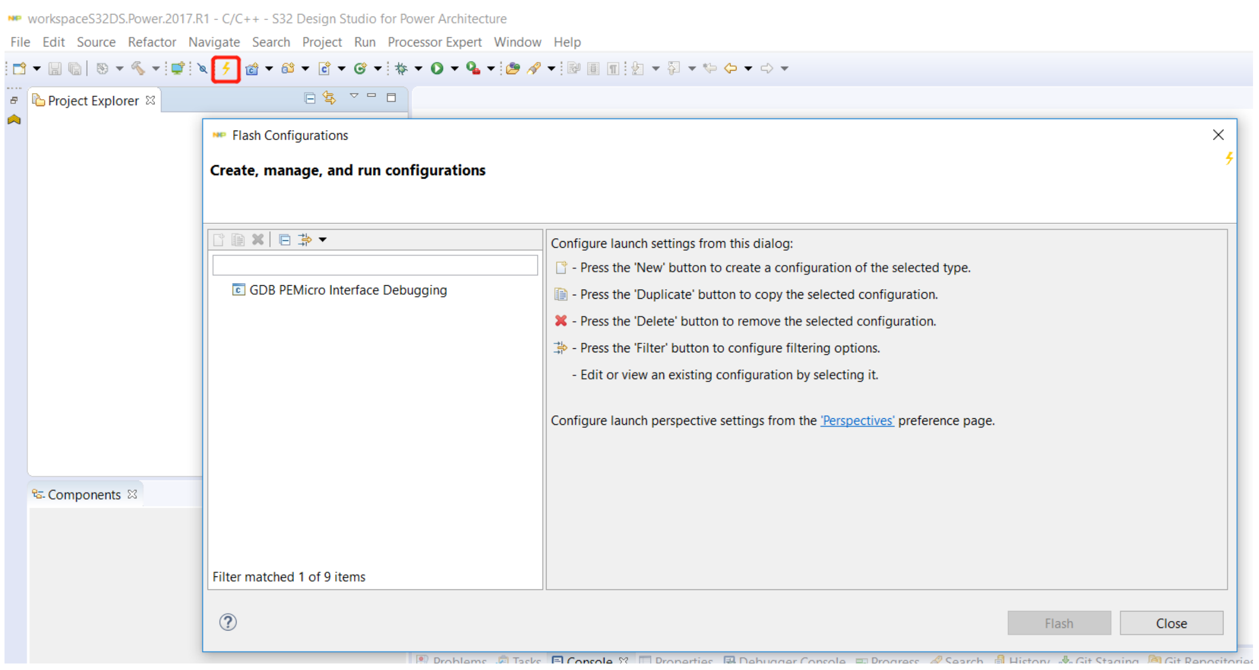
Table 5. Source files

Files	Descriptions
Files with pdc prefix	contain codes for PDC applications.
Files with bcc prefix	contain drivers of MC3377x, include MC33772, MC33771B and MC33771C.
Files with sbc prefix	contain drivers of MC33FS65xx.
Files with boot prefix	contain codes for bootloader.
Files with lin prefix	contain codes for LIN driver implement since SDK has no LIN driver yet.
Files with i2c prefix	contain codes for I^{2}C protocol.
Files with rtc prefix	contain codes for PCA85063 driver.
Files with msdi prefix	contain drivers of CD1020, CD1030, MC33978 and MC34978.
Files with nxp prefix	contain nxp_helpers codes used in SBC drivers.

3.1.3 Downloading exist elf file

The following steps must be followed to download the exist elf file to MCU using S32DS.

1. Launch the S32DS IDE, click on the Flash icon, the Flash Configurations window appears.


workspaceS32DS.Power.2017.R1 - C/C++ - S32 Design Studio for Power Architecture
File Edit Source Refactor Navigate Search Project Run Processor Expert Window Help
Project Explorer
Components
Flash Configurations
GDB PEMicro Interface Debugging
Flash
Close

Create, manage, and run configurations

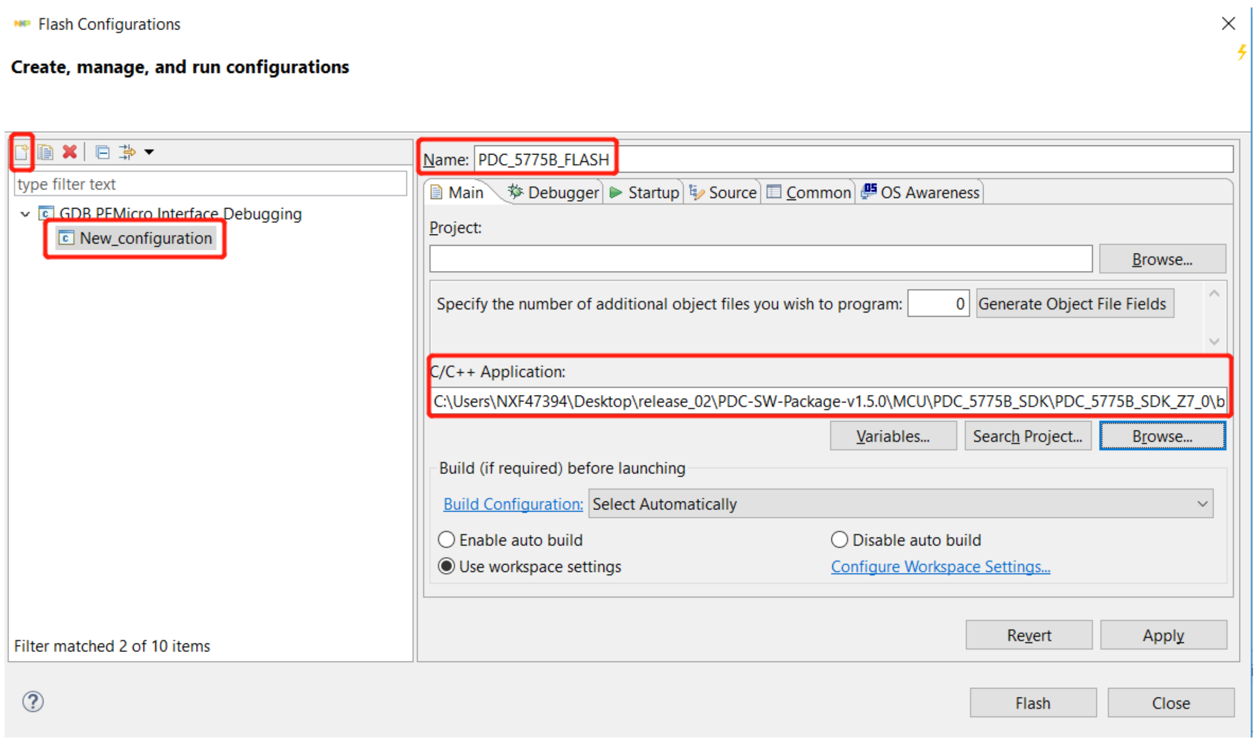
Configure launch settings from this dialog:
- Press the 'New' button to create a configuration of the selected type.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.

Configure launch perspective settings from the 'Perspectives' preference page.

Filter matched 1 of 9 items

Figure 28. Flash configurations

2. Create a new configuration, adjust its name and browse for elf/srec/hex file.



Create, manage, and run configurations

Name: PDC_5775B_FLASH

Project:

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

C/C++ Application:
C:\Users\NXF47394\Desktop\release_02\PDC-SW-Package-v1.5.0\MCU\PDC_5775B_SDK\PDC_5775B_SDK_Z7_0\b

Build (if required) before launching

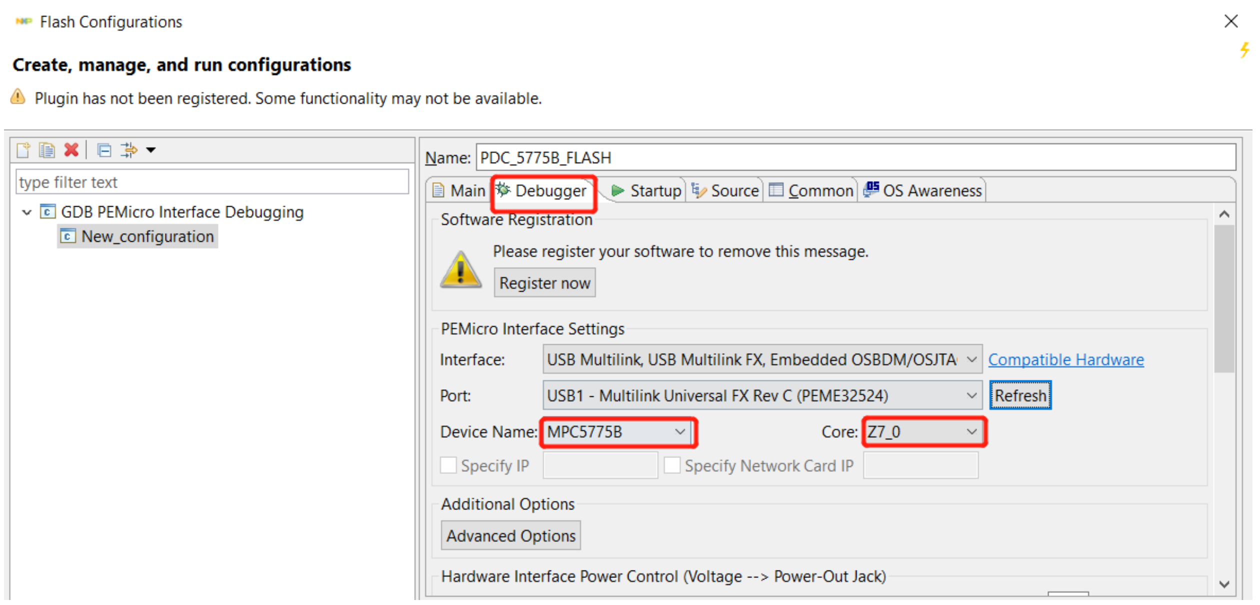
Build Configuration: Select Automatically

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

Filter matched 2 of 10 items

Figure 29. Create new configuration

3. Select MCU and specific core you are targeting.



Name: PDC_5775B_FLASH

Software Registration

Please register your software to remove this message.

PEMicro Interface Settings

Interface: USB Multilink, USB Multilink FX, Embedded OSBDM/OSJTA

Port: USB1 - Multilink Universal FX Rev C (PEME32524)

Device Name: MPC5775B

Core: Z7_0

Specify IP

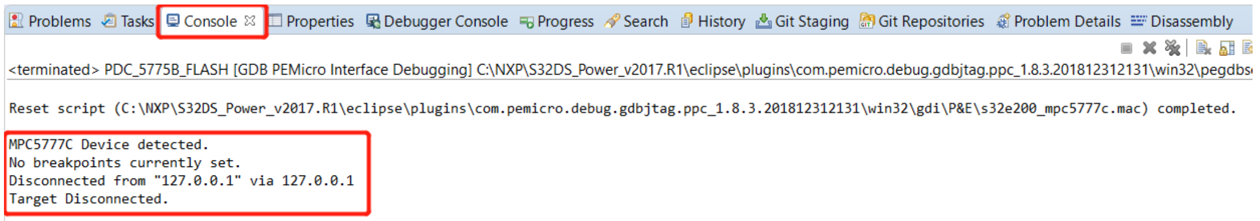
Specify Network Card IP

Additional Options

Hardware Interface Power Control (Voltage --> Power-Out Jack)

Figure 30. Select MCU and specific core

4. Click on Flash button to proceed to programming. As soon as the device is programmed it gets disconnected.



```
<terminated> PDC_5775B_FLASH [GDB PEMicro Interface Debugging] C:\NXP\S32DS_Power_v2017.R1\eclipse\plugins\com.pemicro.debug.gdbjtag.ppc_1.8.3.201812312131\win32\pegdbs

Reset script (C:\NXP\S32DS_Power_v2017.R1\eclipse\plugins\com.pemicro.debug.gdbjtag.ppc_1.8.3.201812312131\win32\gdi\P&E\s32e200_mpc5777c.mac) completed.

MPC5777C Device detected.
No breakpoints currently set.
Disconnected from "127.0.0.1" via 127.0.0.1
Target Disconnected.
```

Figure 31. Flash programming done

3.1.4 Basic workflow of demo project

MCU Init initializes system clock, pins, eDMA and UART. Then it starts to create BSP task, initialize the remaining modules in BSP task, run BSP_TestOnce() function for some module that need to run only once. Finally it creates other tasks that system requires. Start the RTOS scheduler.

SDK project uses FreeRTOS to implement multitasking. The project mainly has four tasks VCU, BMS, BSP and UDS. Software timer is used to trigger BMS (50ms), VCU(10ms) , UDS(100ms) and BSP(300ms) tasks. The callback function of the system software timer is called every 10ms, and schedules multitasking in the timer callback.

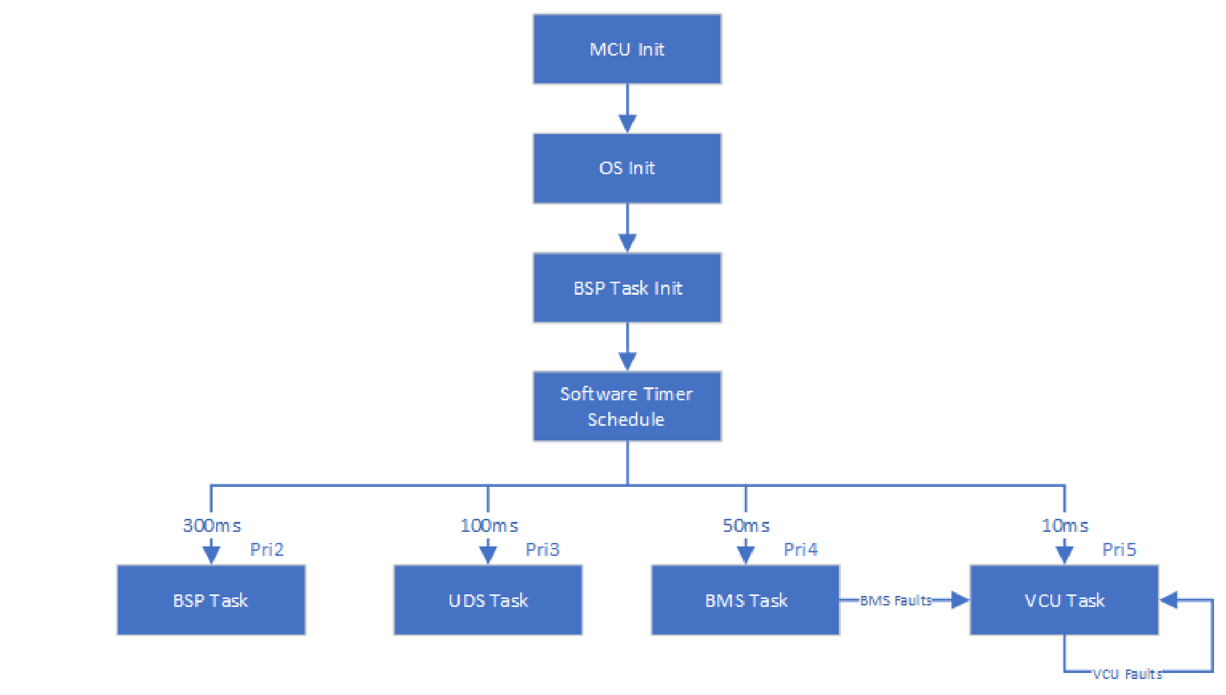


Figure 32. SDK demo project programming flow diagram

- At power up, the SBC watchdog should properly be refreshed during the first 256ms open window.
- VCU task checks gear position, accelerator pedal position, brake pedal position and VCU faults information. Then it calculates the expected output torque. It determines the output, that is, reduce torque and shut down some accessories according to system fault level.
- BMS task monitors stack voltage, cell voltage, temperature and BMS fault information. Support daisy chain of MC33771C or MC33771B or MC33772.
- UDS task monitors specific messages on CAN bus once every 100ms. Once having received commands on CAN bus, UDS Task will execute UDS protocol and jump to bootloader.

NOTE

The default GCC optimization is O2 in the demo project.

3.2 MCAL project

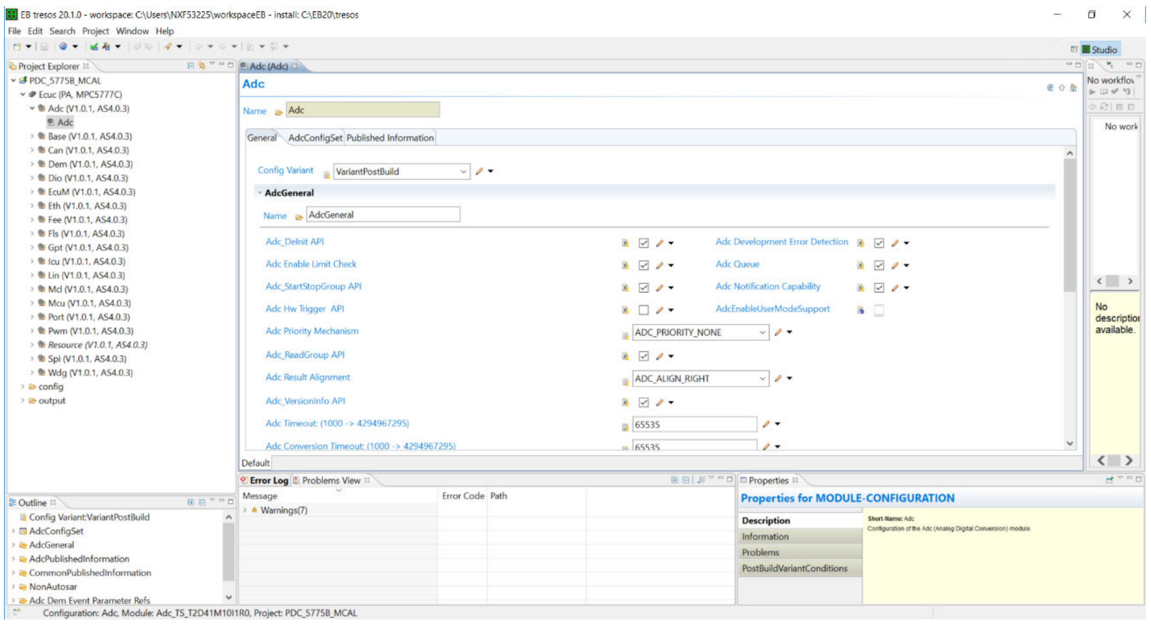
3.2.1 Software and tools required

The following software and tools are required for MCAL project.

- Configuration tool: EB tresos Studio 20.1.0.
- MCAL Drivers: MPC5777C_MCAL4_0_RTM_HF4_1_0_1.
- Compiler: Green Hills Compiler 2018.5.
- Debugger: Lauterbach TRACE32 JTAG Debugger.
- CAN Adaptor: PCAN-USB Pro.

NOTE

There is an issue in MCAL4.0. Please replace *Adc_Eqadc.c* with the provided one in fix folder to enable ADC calibration functions.



```
PDC_5775B_MCAL
  Ecuc (PA, MPC5777C)
    Adc (V1.0.1, AS4.0.3)
      Adc
    Base (V1.0.1, AS4.0.3)
    Can (V1.0.1, AS4.0.3)
    Dem (V1.0.1, AS4.0.3)
    Dio (V1.0.1, AS4.0.3)
    EcuM (V1.0.1, AS4.0.3)
    Eth (V1.0.1, AS4.0.3)
    Fee (V1.0.1, AS4.0.3)
    Fls (V1.0.1, AS4.0.3)
    Gpt (V1.0.1, AS4.0.3)
    Icu (V1.0.1, AS4.0.3)
    Lin (V1.0.1, AS4.0.3)
    Mcl (V1.0.1, AS4.0.3)
    Mcu (V1.0.1, AS4.0.3)
    Port (V1.0.1, AS4.0.3)
    Pwm (V1.0.1, AS4.0.3)
    Resource (V1.0.1, AS4.0.3)
    Spi (V1.0.1, AS4.0.3)
    Wdg (V1.0.1, AS4.0.3)
  config
  output
```

Adc

Name: Adc

Config Variant: VariantPostBuild

AdcGeneral

Name: AdcGeneral

Adc_DeInit API	✓	Adc Development Error Detection	✓
Adc Enable Limit Check	✓	Adc Queue	✓
Adc_StartStopGroup API	✓	Adc Notification Capability	✓
Adc Hw Trigger API	☐	AdcEnableUserModeSupport	☐
Adc Priority Mechanism	ADC_PRIORITY_NONE		
Adc_ReadGroup API	✓		
Adc Result Alignment	ADC_ALIGN_RIGHT		
Adc_VersionInfo API	✓		
Adc Timeout: (1000 -> 4294967295)	65535		
Adc Conversion Timeout: (1000 -> 4294967295)	65535		

Message	Error Code	Path
Warnings(7)		

```
Config Variant:VariantPostBuild
AdcConfigSet
AdcGeneral
AdcPublishedInformation
CommonPublishedInformation
NonAutosar
Adc Dem Event Parameter Refs
```

Properties for MODULE-CONFIGURATION

Description | Information | Problems | PostBuildVariantConditions

Short-Name: Adc
Configuration of the Adc (Analog Digital Conversion) module

No description available.

Configuration: Adc, Module: Adc_TS_T2D41M10I1R0, Project: PDC_5775B_MCAL

Figure 33. EB tresos

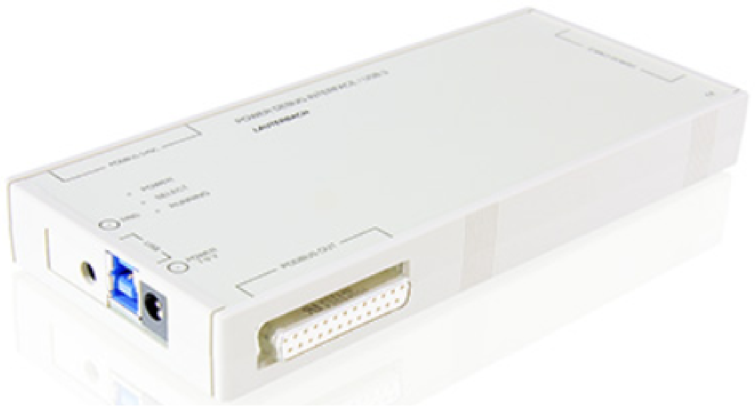


Figure 34. Lauterbach debugger

Figure 35. PCAN-USB pro

3.2.2 File structure

Table 6. MCAL project files

Folders/Files	Descriptions
.metadata	contains log information of Tresos.
bin	contains generated object files.
bin/bootloader_bin	contains bootloader binary files.
cfg	contains configuration files generated by Tresos.
fix	contains files to fix bugs.
include	contains application header files with pdc_ prefix.
make	contains makefiles used for building the application.
src	contains application source code files.
toolchains	contains files needed to build with various toolchains (startup, linker command files).
Tresos	contains the Tresos project with the application configuration.
makefile	is the MCAL sample application makefile.

Table continues on the next page...

Table 6. MCAL project files (continued)

Folders/Files	Descriptions
Modules	specifies which modules are compiled and linked.
make.bat	launches the make command.
launch.bat	contains user-defined paths for compilation and launches the make.bat file.
clear.bat	clears all generated and built files.
compiler_used.s	specifies which complier is used.
run.cmm	is a script used for Lauterbach debugger

NOTE

For bootloader supported project, please unpack “Bootloader_Settings.zip” in toolchains older to replace *autosar_flash.ld*, *Startup_vle.c* and *Vector_vle_core.s*. If you do not want a bootloader, unpack Non_Bootloader_Settings.zip.

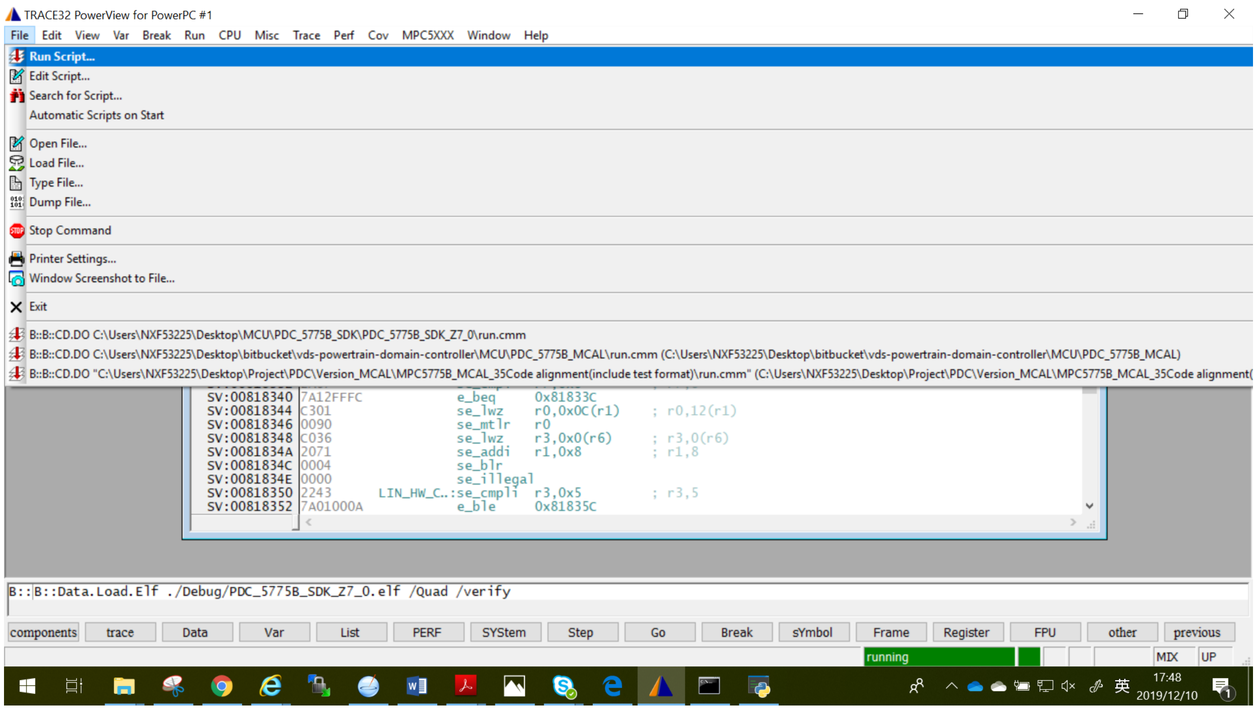
Table 7. Source files

Files	Descriptions
Files with pdc prefix	contain codes for PDC applications.
Files with bcc prefix	contain drivers of MC3377x, include MC33772, MC33771B and MC33771C.
Files with sbc prefix	contain drivers of MC33FS65xx.
Files in boot folder	contain codes for bootloader.
Files with i2c prefix	contain codes for I2C protocol.
Files with rtc prefix	contain codes for PCA85063 driver.
Files with msdi prefix	contain drivers of CD1020, CD1030, MC33978 and MC34978.
Files with nxp prefix	contain nxp_helpers codes used in SBC drivers.

3.2.3 Downloading the exist elf file

For downloading the exist elf file using TRACE32 follow these stps.

1. Launch the TRACE32. Select File > Run Script…, from the menu bar.


TRACE32 PowerView for PowerPC #1
File Edit View Var Break Run CPU Misc Trace Perf Cov MPC5XXX Window Help
Run Script...
Edit Script...
Search for Script...
Automatic Scripts on Start
Open File...
Load File...
Type File...
Dump File...
Stop Command
Printer Settings...
Window Screenshot to File...
Exit
components trace Data Var List PERF SYStem Step Go Break sYmbol Frame Register FPU other previous
running
MIX UP
17:48
2019/12/10

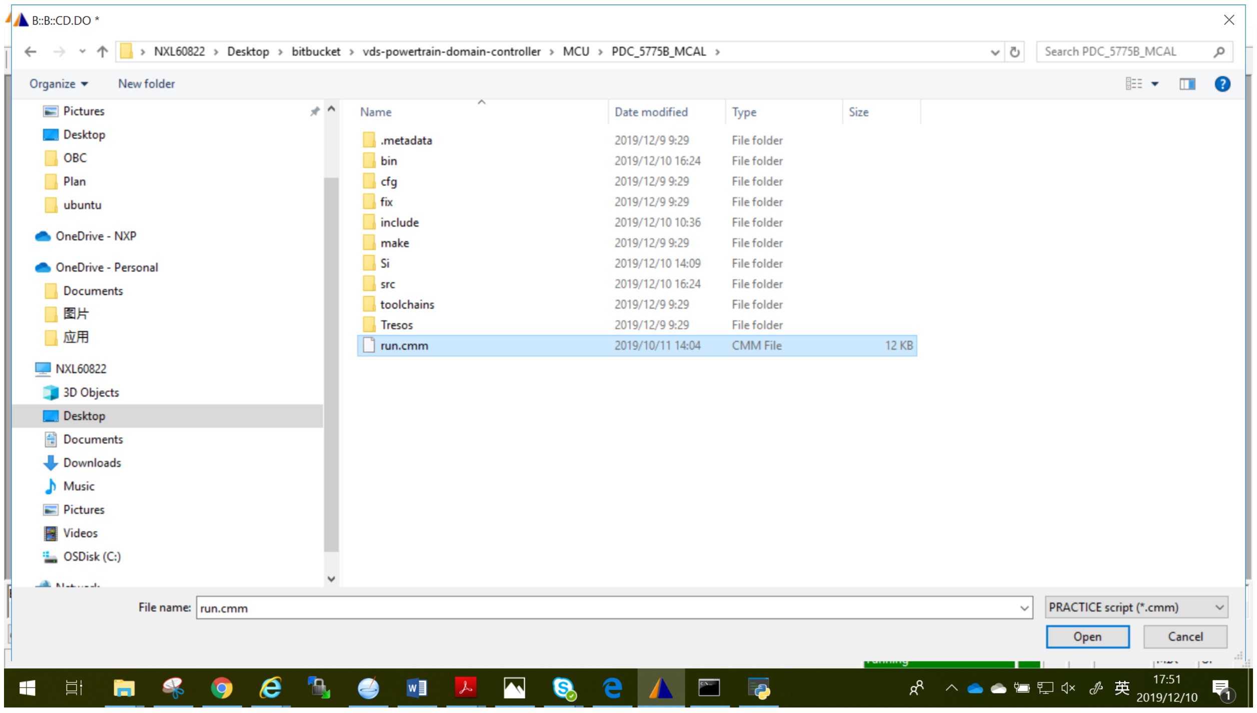
```
B::B::CD.DO C:\Users\NXF53225\Desktop\MCU\PDC_5775B_SDK\PDC_5775B_SDK_Z7_0\run.cmm
B::B::CD.DO C:\Users\NXF53225\Desktop\bitbucket\vds-powertrain-domain-controller\MCU\PDC_5775B_MCAL\run.cmm (C:\Users\NXF53225\Desktop\bitbucket\vds-powertrain-domain-controller\MCU\PDC_5775B_MCAL)
B::B::CD.DO "C:\Users\NXF53225\Desktop\Project\PDC\Version_MCAL\MPC5775B_MCAL_35Code alignment(include test format)\run.cmm" (C:\Users\NXF53225\Desktop\Project\PDC\Version_MCAL\MPC5775B_MCAL_35Code alignment(
```

```
SV:00818340 7A12FFFC             e_beq     0x81833C
SV:00818344 C301                 se_lwz    r0,0x0C(r1)     ; r0,12(r1)
SV:00818346 0090                 se_mtlr   r0
SV:00818348 C036                 se_lwz    r3,0x0(r6)      ; r3,0(r6)
SV:0081834A 2071                 se_addi   r1,0x8          ; r1,8
SV:0081834C 0004                 se_blr
SV:0081834E 0000                 se_illegal
SV:00818350 2243      LIN_HW_C..:se_cmpli  r3,0x5          ; r3,5
SV:00818352 7A01000A             e_ble     0x81835C
```

```
B::B::Data.Load.Elf ./Debug/PDC_5775B_SDK_Z7_0.elf /Quad /verify
```

Figure 36. Run script

2. Select the script named run.cmm (in root directory) to download elf. file into flash.

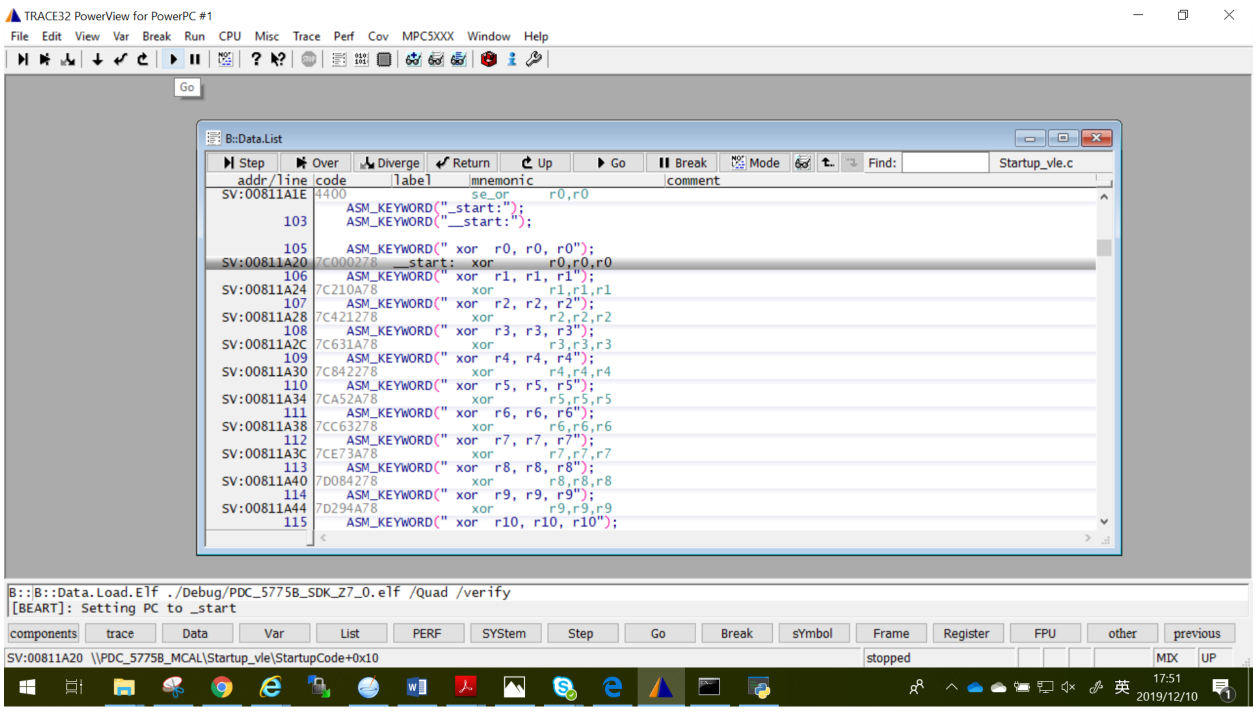


Name	Date modified	Type	Size
.metadata	2019/12/9 9:29	File folder	
bin	2019/12/10 16:24	File folder	
cfg	2019/12/9 9:29	File folder	
fix	2019/12/9 9:29	File folder	
include	2019/12/10 10:36	File folder	
make	2019/12/9 9:29	File folder	
Si	2019/12/10 14:09	File folder	
src	2019/12/10 16:24	File folder	
toolchains	2019/12/9 9:29	File folder	
Tresos	2019/12/9 9:29	File folder	
run.cmm	2019/10/11 14:04	CMM File	12 KB

File name: run.cmm

Figure 37. Browse script file

3. Click Go button to run.


TRACE32 PowerView for PowerPC #1
File Edit View Var Break Run CPU Misc Trace Perf Cov MPC5XXX Window Help
Go
B::Data.List
Step Over Diverge Return Up Go Break Mode Find: Startup_vle.c
components trace Data Var List PERF SYStem Step Go Break sYmbol Frame Register FPU other previous
stopped MIX UP
17:51 2019/12/10

```
      addr/line code      label     mnemonic              comment
SV:00811A1E 4400                se_or     r0,r0
                ASM_KEYWORD("_start:");
        103     ASM_KEYWORD("__start:");

        105     ASM_KEYWORD(" xor  r0, r0, r0");
SV:00811A20 7C000278  __start:  xor       r0,r0,r0
        106     ASM_KEYWORD(" xor  r1, r1, r1");
SV:00811A24 7C210A78            xor       r1,r1,r1
        107     ASM_KEYWORD(" xor  r2, r2, r2");
SV:00811A28 7C421278            xor       r2,r2,r2
        108     ASM_KEYWORD(" xor  r3, r3, r3");
SV:00811A2C 7C631A78            xor       r3,r3,r3
        109     ASM_KEYWORD(" xor  r4, r4, r4");
SV:00811A30 7C842278            xor       r4,r4,r4
        110     ASM_KEYWORD(" xor  r5, r5, r5");
SV:00811A34 7CA52A78            xor       r5,r5,r5
        111     ASM_KEYWORD(" xor  r6, r6, r6");
SV:00811A38 7CC63278            xor       r6,r6,r6
        112     ASM_KEYWORD(" xor  r7, r7, r7");
SV:00811A3C 7CE73A78            xor       r7,r7,r7
        113     ASM_KEYWORD(" xor  r8, r8, r8");
SV:00811A40 7D084278            xor       r8,r8,r8
        114     ASM_KEYWORD(" xor  r9, r9, r9");
SV:00811A44 7D294A78            xor       r9,r9,r9
        115     ASM_KEYWORD(" xor  r10, r10, r10");
```

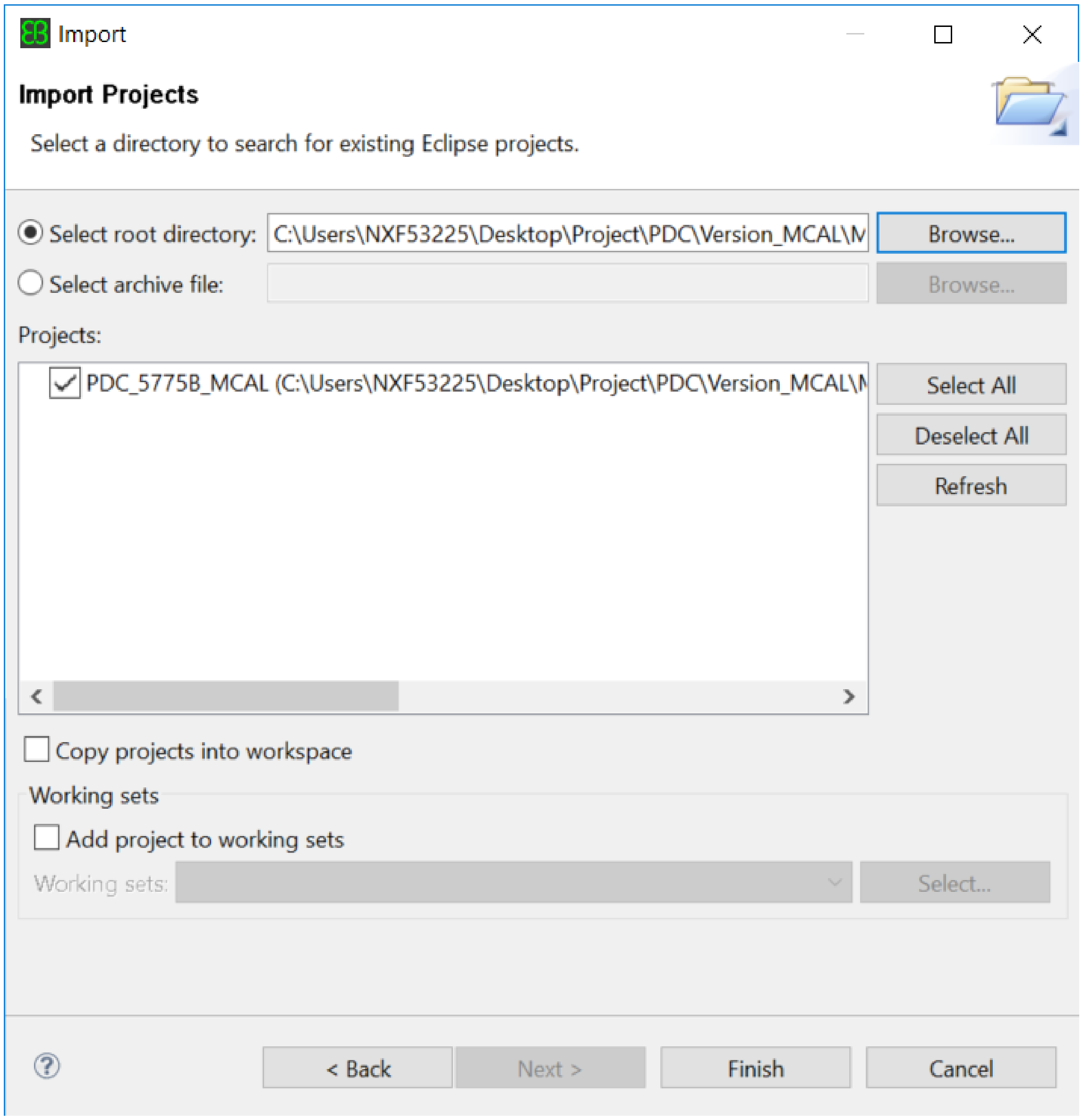
```
B::B::Data.Load.Elf ./Debug/PDC_5775B_SDK_Z7_0.elf /Quad /verify
[BEART]: Setting PC to _start
```

```
SV:00811A20  \\PDC_5775B_MCAL\Startup_vle\StartupCode+0x10
```

Figure 38. Programming running

3.2.4 How to build

1. Launch the EB tresos Studio and select File > Import, from the IDE menu. Choose import source page appears. Click Browse and select the project folder \Tresos\workspace, click Finish.


Import

Import Projects

Select a directory to search for existing Eclipse projects.

(●) Select root directory: C:\Users\NXF53225\Desktop\Project\PDC\Version_MCAL\M [Browse...]

( ) Select archive file: [Browse...]

Projects:

☑ PDC_5775B_MCAL (C:\Users\NXF53225\Desktop\Project\PDC\Version_MCAL\M

☐ Copy projects into workspace

Working sets

☐ Add project to working sets

Working sets: [Select...]

Figure 39. Import projects

2. Double click the project name PDC_5775B_MCAL and click generate button.

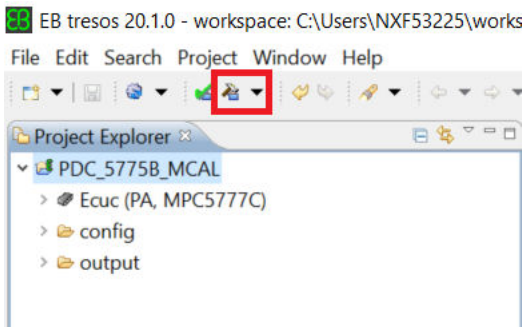

EB tresos 20.1.0 - workspace: C:\Users\NXF53225\works

Figure 40. Generate code

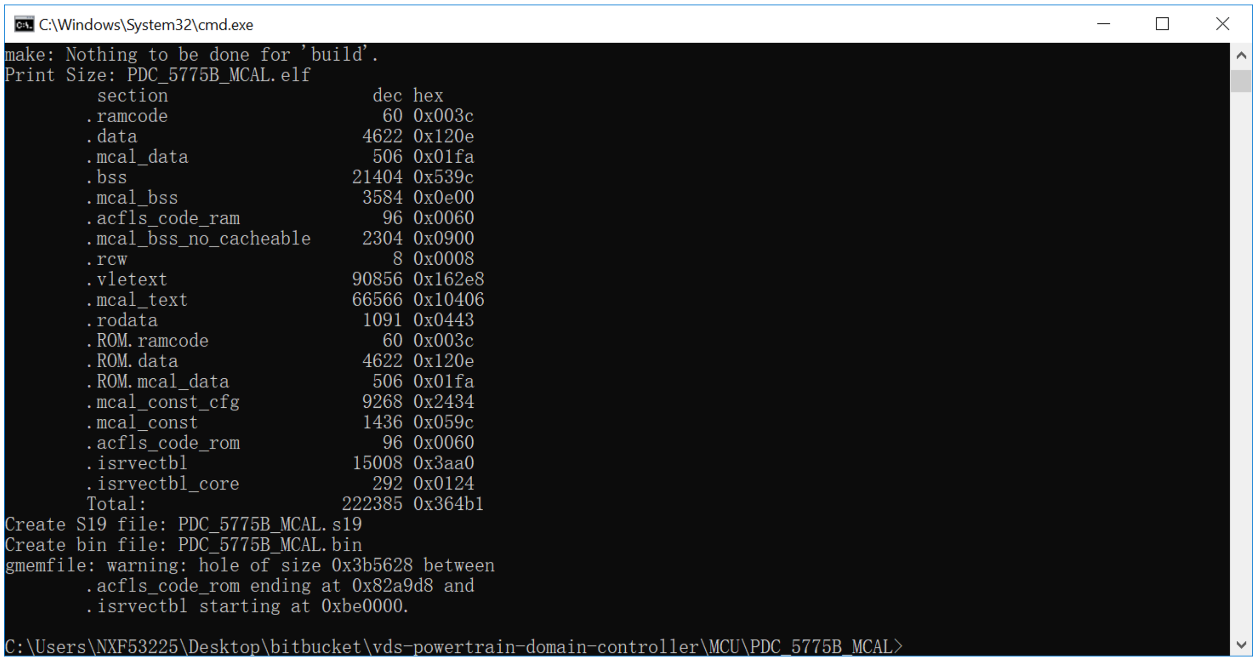
3. Fill in the right compiler in compiler_used.s.
4. Fill in user-defined paths for compilation in launch.bat. Paths includes TRESOS_DIR, MAKE_DIR, GHS_DIR or DIAB_DIR or IAR_DIR, PLUGINS_DIR.

```
@echo off

::You have to uncomment and set the following variables if they are not already set
:: uncomment line below if you do not set TRESOS_DIR over environment
::TRESOS
SET TRESOS_DIR=C:\EB20\tresos
::MAKE
SET MAKE_DIR=C:\NXP\S32DS_Power_v2017.R1\utils\msys32\usr
::GHS
SET GHS_DIR=C:\ghs\comp_201854
::DIAB
::SET DIAB_DIR=C:/Tools/WindRiver/diab/5.9.4.7
::IAR
::SET IAR_DIR=C:/Tools/IAR Systems/Embedded Workbench 7.3/arm/
::Path to the plugins folder
SET PLUGINS_DIR=C:\NXP\AUTOSAR\MPC5777C_MCAL4_0_RTM_HF4_1_0_1\eclipse\plugins
::SSC
::SET SSC_ROOT=
```

Figure 41. Lauch.bat file

5. Run build.bat in command window. Errors, warnings and compilation information can be check here.


C:\Windows\System32\cmd.exe

```
make: Nothing to be done for 'build'.
Print Size: PDC_5775B_MCAL.elf
         section                    dec hex
        .ramcode                     60 0x003c
        .data                      4622 0x120e
        .mcal_data                  506 0x01fa
        .bss                      21404 0x539c
        .mcal_bss                  3584 0x0e00
        .acfls_code_ram              96 0x0060
        .mcal_bss_no_cacheable     2304 0x0900
        .rcw                          8 0x0008
        .vletext                  90856 0x162e8
        .mcal_text                66566 0x10406
        .rodata                    1091 0x0443
        .ROM.ramcode                 60 0x003c
        .ROM.data                  4622 0x120e
        .ROM.mcal_data              506 0x01fa
        .mcal_const_cfg            9268 0x2434
        .mcal_const                1436 0x059c
        .acfls_code_rom              96 0x0060
        .isrvectbl                15008 0x3aa0
        .isrvectbl_core             292 0x0124
        Total:                   222385 0x364b1
Create S19 file: PDC_5775B_MCAL.s19
Create bin file: PDC_5775B_MCAL.bin
gmemfile: warning: hole of size 0x3b5628 between
        .acfls_code_rom ending at 0x82a9d8 and
        .isrvectbl starting at 0xbe0000.

C:\Users\NXF53225\Desktop\bitbucket\vds-powertrain-domain-controller\MCU\PDC_5775B_MCAL>
```

Figure 42. Build result

6. You can run clear.bat to clear output objects in unexpected situations and build again.

NOTE

Recommended optimization option for this demo is -Osize if Green Hills compiler is used.

3.2.5 Basic workflow of memo project

MCU Init is executed first. It initializes system clock, external interrupts, ports and UART.

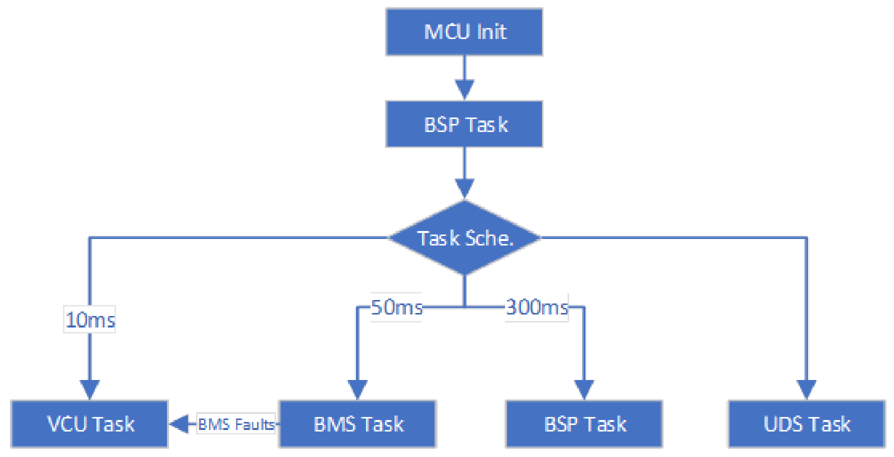


Figure 43. MCAL demo project programming flow diagram

BSP task takes four responsibilities:

1. BSP Task initializes the remaining peripherals after MCU Init, including DSPI used for initializing SBC watchdog.
2. BSP Task initializes SBC watchdog which needs to be initialized within at most 256ms from power on. This is why SBC Task shall be executed before starting task scheduler.
3. BSP Task refreshes SBC watchdog regularly after initialization.
4. BSP Task contains test functions which shall be executed once or regularly.

Task Scheduler calls three tasks cyclically. The calling periods are 10ms, 50ms and 300ms respectively for VCU Task, BMS Task and BSP Task. UDS Task runs freely instead of scheduled by Task Scheduler because UDS has strict timing requirements and there is no OS to guarantee the schedule can meet UDS timing requirements. Once having received commands on CAN bus, UDS Task will execute UDS protocol and jump to bootloader.

VCU Task checks gear position, accelerator & brake pedal position and VCU faults. After considering BMS faults and VCU faults, VCU Task controls the output, that is, reduces output torque and shuts done accessories by LSD.

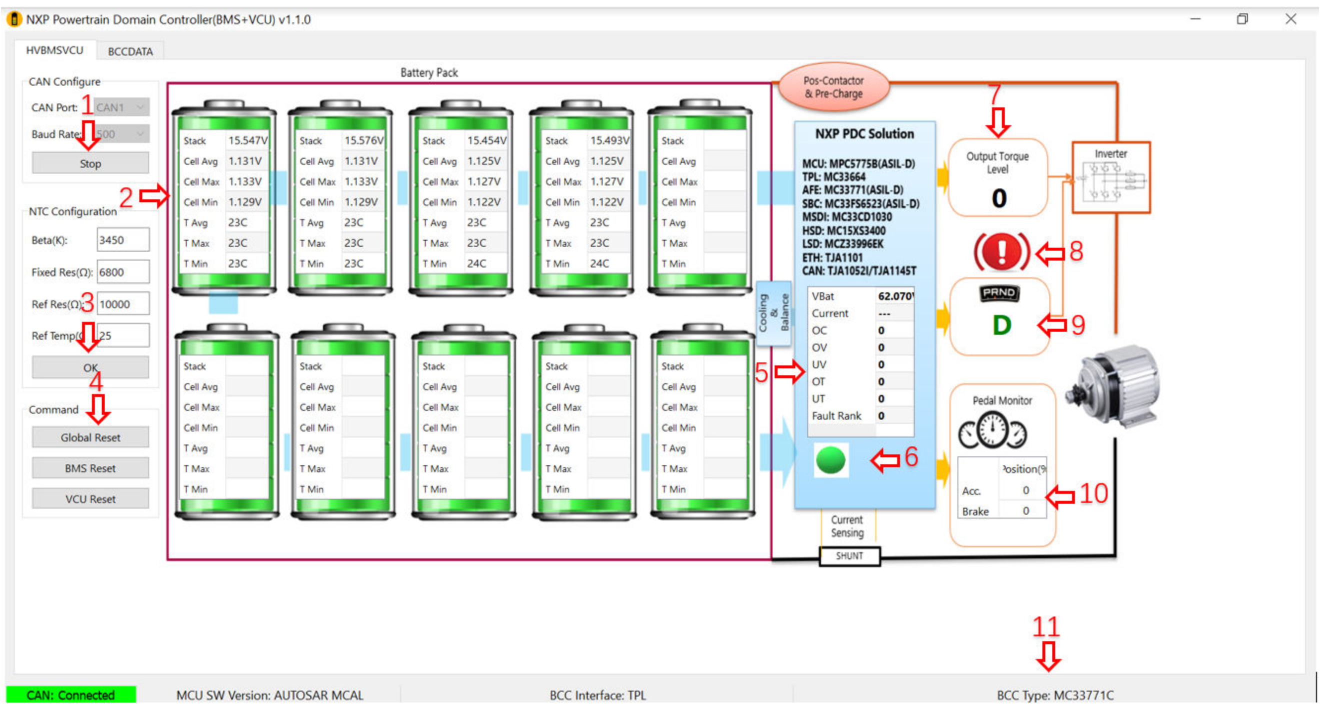
BMS Task monitors stack voltage, cell voltage, temperature and BMS fault information. Support daisy chain of MC33771C or MC33771B or MC33772.

NOTE

1. There is no OS implemented in MCAL project to schedule tasks, one task will only be executed after the previous one is finished.
2. There is a limitation in ETH module. You need to install ETH_RxCbk callback into plugin codes to enable ETH receive function.
   a. Copy the following code into EthIf_RxIndication ETH_RxCbk(CtrlIdx, FrameType, IsBroadcast, PhysAddrPtr, DataPtr, LenByte)
   a. Include pdc_eth.h in EthIf_Cbk.c.

Chapter 4
GUI user guide

This GUI tool is used to monitor system via CAN bus, which can display the system state real time and send reset command to board. It need to use PCAN-USB Pro adapter to connect board with PC. Hands-on instructions are listed below.


NXP Powertrain Domain Controller(BMS+VCU) v1.1.0

HVBMSVCU | BCCDATA

CAN Configure
CAN Port: CAN1
Baud Rate: 500

NTC Configuration
Beta(K): 3450
Fixed Res(Ω): 6800
Ref Res(Ω): 10000
Ref Temp(C): 25

Battery Pack

Stack	15.547V	Stack	15.576V	Stack	15.454V	Stack	15.493V	Stack	
Cell Avg	1.131V	Cell Avg	1.131V	Cell Avg	1.125V	Cell Avg	1.125V	Cell Avg	
Cell Max	1.133V	Cell Max	1.133V	Cell Max	1.127V	Cell Max	1.127V	Cell Max	
Cell Min	1.129V	Cell Min	1.129V	Cell Min	1.122V	Cell Min	1.122V	Cell Min	
T Avg	23C	T Avg	23C	T Avg	23C	T Avg	23C	T Avg	
T Max	23C	T Max	23C	T Max	23C	T Max	23C	T Max	
T Min	23C	T Min	23C	T Min	24C	T Min	24C	T Min	

Stack		Stack		Stack		Stack		Stack	
Cell Avg		Cell Avg		Cell Avg		Cell Avg		Cell Avg	
Cell Max		Cell Max		Cell Max		Cell Max		Cell Max	
Cell Min		Cell Min		Cell Min		Cell Min		Cell Min	
T Avg		T Avg		T Avg		T Avg		T Avg	
T Max		T Max		T Max		T Max		T Max	
T Min		T Min		T Min		T Min		T Min	

NXP PDC Solution
MCU: MPC5775B(ASIL-D)
TPL: MC33664
AFE: MC33771(ASIL-D)
SBC: MC33FS6523(ASIL-D)
MSDI: MC33CD1030
HSD: MC15XS3400
LSD: MCZ33996EK
ETH: TJA1101
CAN: TJA1052I/TJA1145T

VBat	62.070V
Current	---
OC	0
OV	0
UV	0
OT	0
UT	0
Fault Rank	0

Output Torque Level
0

D

Pedal Monitor

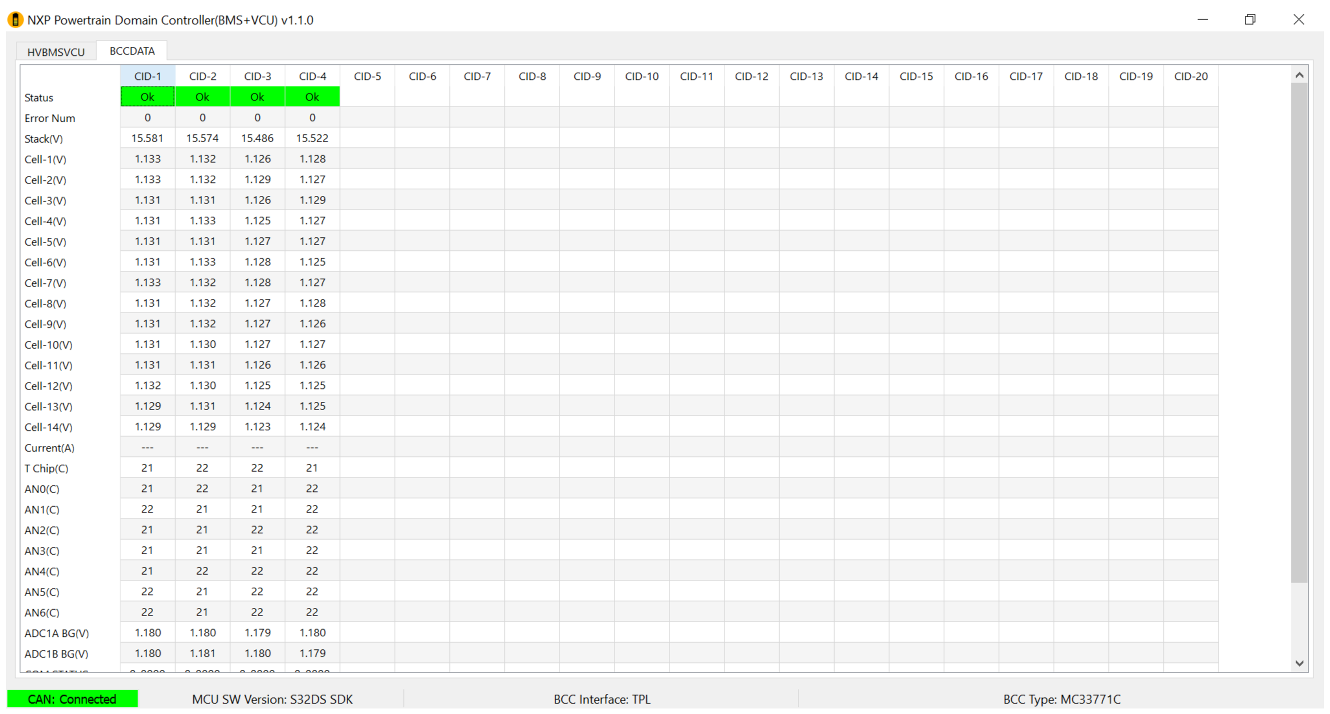
	Position(%)
Acc.	0
Brake	0

CAN: Connected | MCU SW Version: AUTOSAR MCAL | BCC Interface: TPL | BCC Type: MC33771C

Figure 44. GUI layout

1. Select CAN port and Baud rate (500 Kbps for this demo) of CAN transmission and then click Start button. Click Stop button when you want to disconnect the CAN adapter.
2. Battery voltage will be displayed, including stack voltage and cell voltage(average, maximal and minimal). Temperature(average, maximal and minimal) is also displayed. This GUI supports up to 10 battery stacks.
3. NTC parameters can be changed according to populated NTC.
4. Clicking Global Reset button resets BMS and VCU task together. While BMS Reset and VCU Reset button reset BMS and VCU task separately.
5. Total voltage, current (not measured in this demo) and fault information is displayed.
6. This indicator becomes red when faults occur otherwise becomes green.
7. Output torque level (0, Low, Mid and High) is shown.
8. This indicator appears when handbrake is pulled up.
9. Gear position (R, N, and D) is shown. E will appear when the gear shift fails .
10. Positions of accelerate pedal and brake pedal is shown. If measured values are out of range, an error image will appear.
11. Basic system information is shown on status bar.

All BCC data are displayed on the BCCDATA page (the following figure), where the first row indicates the status of the CID, including OK, timeout, CRC, etc. The last four rows indicates BCC communication errors and BCC faults.



NXP Powertrain Domain Controller(BMS+VCU) v1.1.0

HVBMSVCU | BCCDATA

	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-13	CID-14	CID-15	CID-16	CID-17	CID-18	CID-19	CID-20
Status	Ok	Ok	Ok	Ok																
Error Num	0	0	0	0																
Stack(V)	15.581	15.574	15.486	15.522																
Cell-1(V)	1.133	1.132	1.126	1.128																
Cell-2(V)	1.133	1.132	1.129	1.127																
Cell-3(V)	1.131	1.131	1.126	1.129																
Cell-4(V)	1.131	1.133	1.125	1.127																
Cell-5(V)	1.131	1.131	1.127	1.127																
Cell-6(V)	1.131	1.133	1.128	1.125																
Cell-7(V)	1.133	1.132	1.128	1.127																
Cell-8(V)	1.131	1.132	1.127	1.128																
Cell-9(V)	1.131	1.132	1.127	1.126																
Cell-10(V)	1.131	1.130	1.127	1.127																
Cell-11(V)	1.131	1.131	1.126	1.126																
Cell-12(V)	1.132	1.130	1.125	1.125																
Cell-13(V)	1.129	1.131	1.124	1.125																
Cell-14(V)	1.129	1.129	1.123	1.124																
Current(A)	---	---	---	---																
T Chip(C)	21	22	22	21																
AN0(C)	21	22	21	22																
AN1(C)	22	21	21	22																
AN2(C)	21	21	22	22																
AN3(C)	21	21	21	22																
AN4(C)	21	22	22	22																
AN5(C)	22	21	22	22																
AN6(C)	22	21	22	22																
ADC1A BG(V)	1.180	1.180	1.179	1.180																
ADC1B BG(V)	1.180	1.181	1.180	1.179																

CAN: Connected | MCU SW Version: S32DS SDK | BCC Interface: TPL | BCC Type: MC33771C

Figure 45. BCC data page

Chapter 5
Bootloader User Guide

A bootloader is provided to update application firmware via CAN bus in case there are no tools to download application directly. It uses PCAN-USB Pro adapter to connect board with PC, executes its function by OpenBus tool. Hands-on instructions are listed below. Refer to the documentation "Unified bootloader user guide" to understand more details about bootloader source code.

1. Download bootloader into MCU: Download MPC577XX_CAN_bootloader.elf into flash by PE in S32DS or Lauterbach with run.cmm.
2. Allocate flash memory of the application. Unpack Bootloader_Settings.zip to replace original files, then compile the APP project.
3. Launch the OpenBus

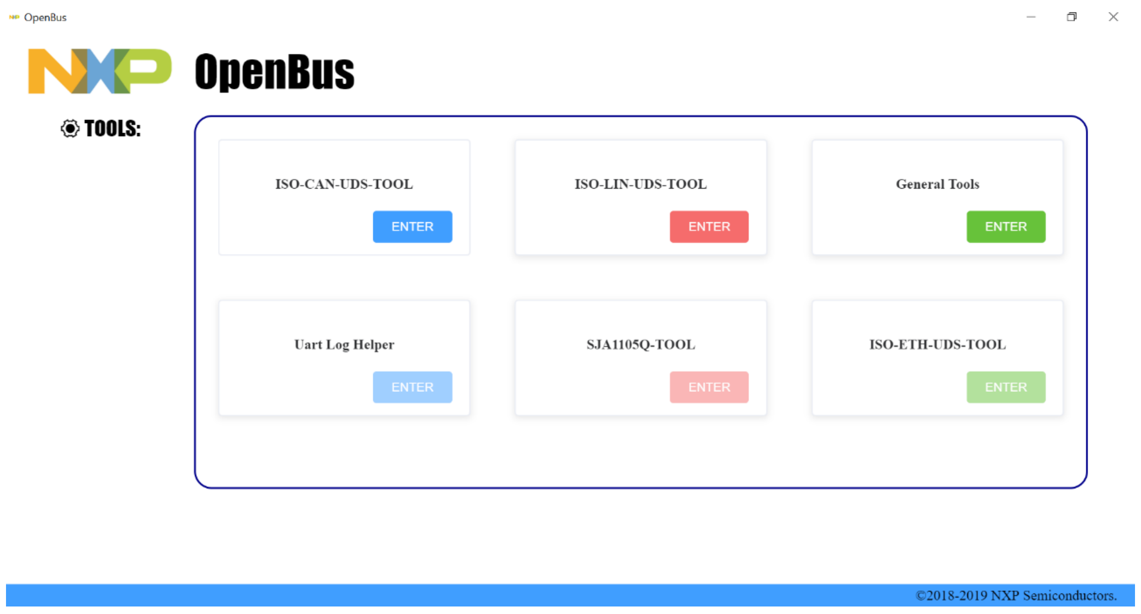
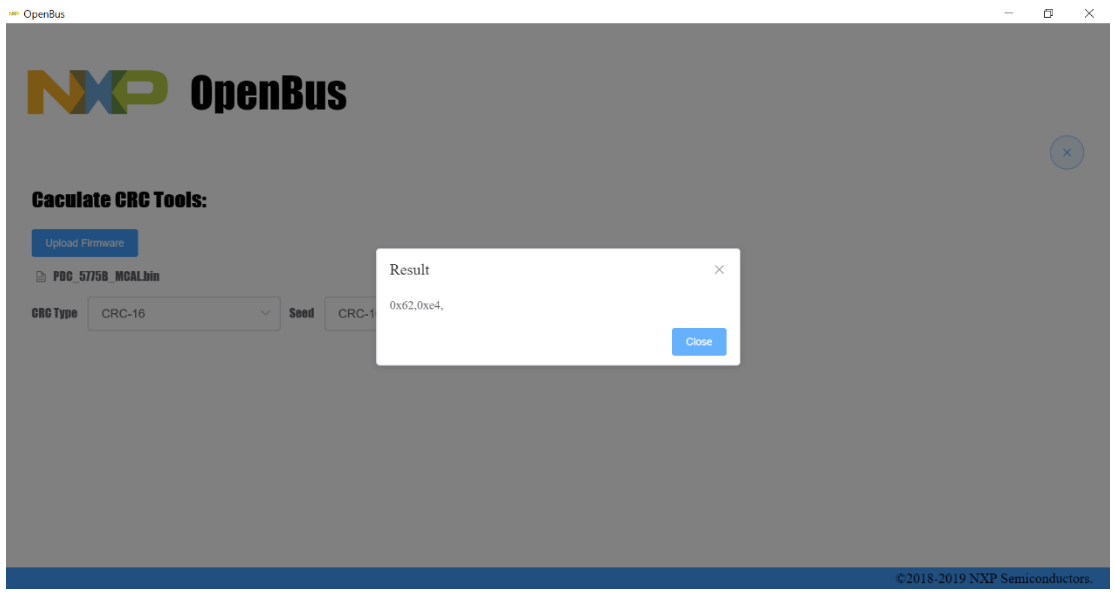


Figure 46. OpenBus tools homepage

4. Calculate CRC for the binary file to be transferred:
   - Click Enter button on General Tools block.
   - Click Upload Firmware button and select the binary file to be transferred.
   - Select CRC-16(DNP) algorithm.
   - Click Calculate button to generate CRC value.

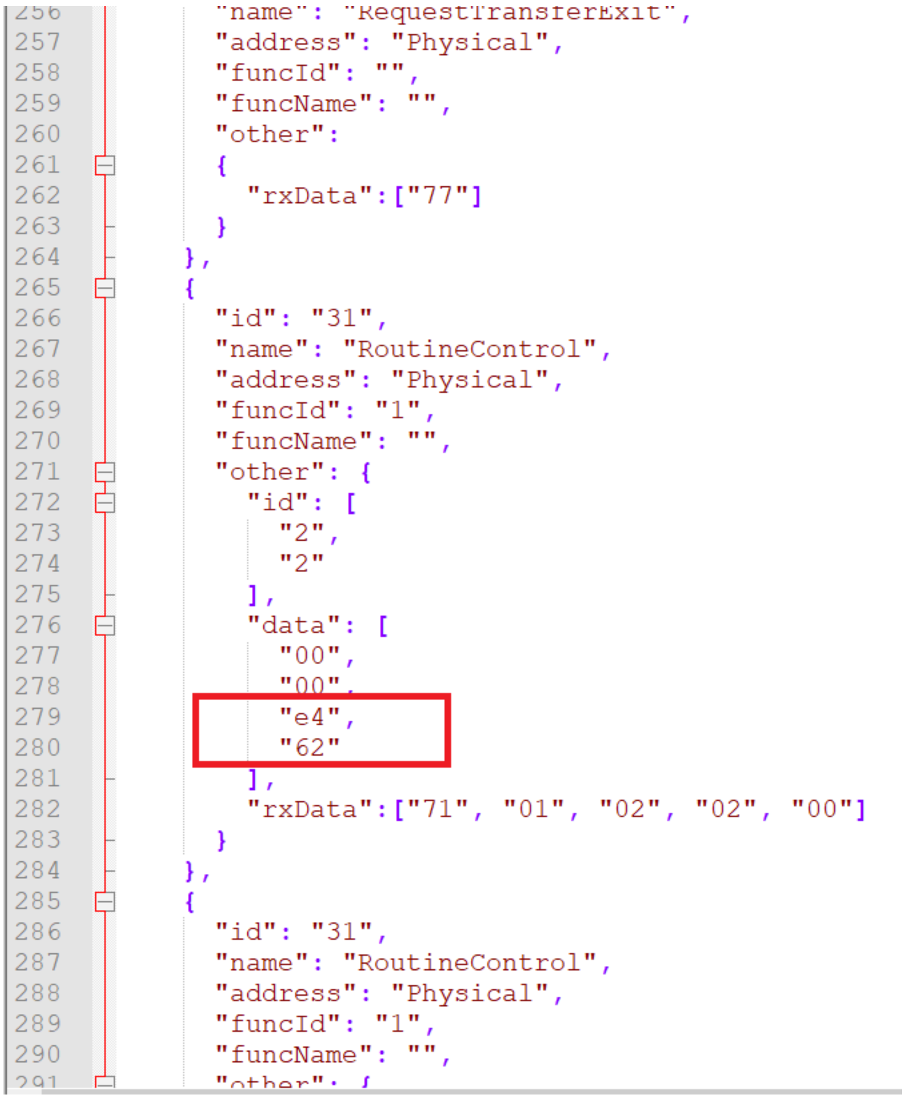


Result

0x62,0xe4,

Figure 47. CRC-16(DNP) example

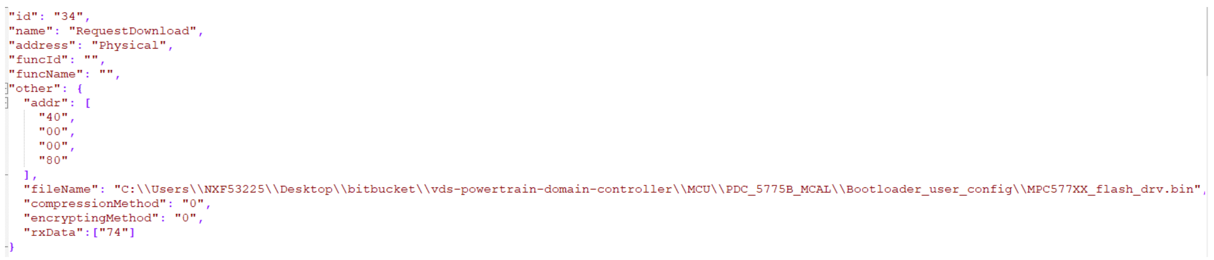
5. Edit configuration file can_uds.json in bootloader_bin folder:
   - Fill in the CRC value calculated in Step 4.



```
        "name": "RequestTransferExit",
        "address": "Physical",
        "funcId": "",
        "funcName": "",
        "other":
        {
          "rxData":["77"]
        }
      },
      {
        "id": "31",
        "name": "RoutineControl",
        "address": "Physical",
        "funcId": "1",
        "funcName": "",
        "other": {
          "id": [
            "2",
            "2"
          ],
          "data": [
            "00",
            "00",
            "e4",
            "62"
          ],
          "rxData":["71", "01", "02", "02", "00"]
        }
      },
      {
        "id": "31",
        "name": "RoutineControl",
        "address": "Physical",
        "funcId": "1",
        "funcName": "",
        "other": {
```

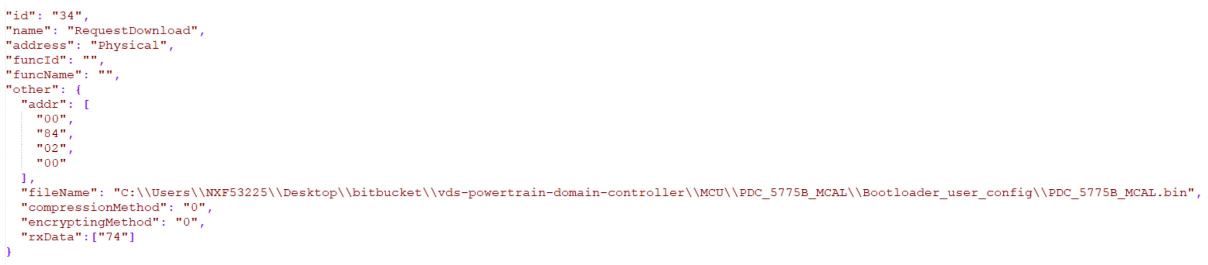
Figure 48. CRC in can_uds.json example

- Fill in the correct paths of MPC577XX_flash_drv.bin and app.bin (here PDC_5775B_MCAL.bin).



```
"id": "34",
"name": "RequestDownload",
"address": "Physical",
"funcId": "",
"funcName": "",
"other": {
  "addr": [
    "40",
    "00",
    "00",
    "80"
  ],
  "fileName": "C:\\Users\\NXF53225\\Desktop\\bitbucket\\vds-powertrain-domain-controller\\MCU\\PDC_5775B_MCAL\\Bootloader_user_config\\MPC577XX_flash_drv.bin",
  "compressionMethod": "0",
  "encryptingMethod": "0",
  "rxData":["74"]
}
```

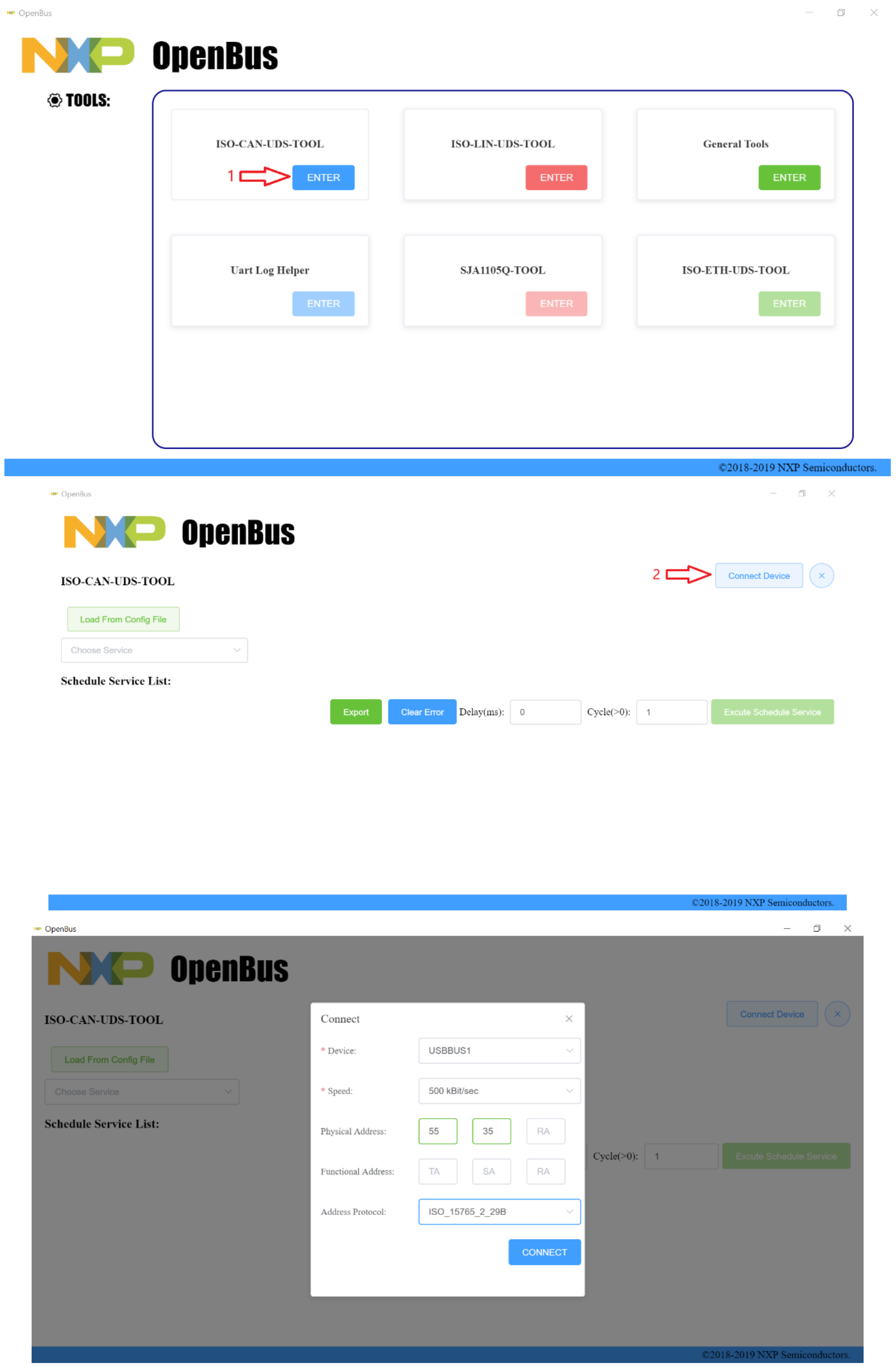
Figure 49. Path of MPC577XX_flash_drv.bin example



```
"id": "34",
"name": "RequestDownload",
"address": "Physical",
"funcId": "",
"funcName": "",
"other": {
  "addr": [
    "00",
    "84",
    "02",
    "00"
  ],
  "fileName": "C:\\Users\\NXF53225\\Desktop\\bitbucket\\vds-powertrain-domain-controller\\MCU\\PDC_5775B_MCAL\\Bootloader_user_config\\PDC_5775B_MCAL.bin",
  "compressionMethod": "0",
  "encryptingMethod": "0",
  "rxData":["74"]
}
```

Figure 50. Path of PDC_5775B_MCAL.bin example

6. Connect PCAN-USB pro:
   - Click Enter button on ISO-CAN-UDS-TOOL block.
   - Click Connect Device on the right-top.
   - Select and fill values as shown below and then click Connect button. Note that target address and source address are fixed as 0x55 and 0x35.



Delay(ms): 0 Cycle(>0): 1

Connect

* Device:	USBBUS1
* Speed:	500 kBit/sec
Physical Address:	55 · 35 · RA
Functional Address:	TA · SA · RA
Address Protocol:	ISO_15765_2_29B

Figure 51. Parameters for connecting device

7. Load config files which links the target binary files. Click Load From Config File button and select can_uds.json you just modified in step 5.

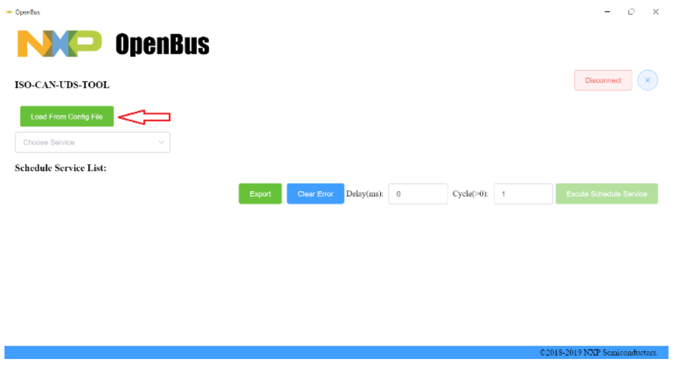


Figure 52. Load from config file

8. Execute bootloader procedure:
   - Click Execute Schedule Service button to execute bootloader procedure.
   - Wait the process to succeed.
   - You can repeat Step 4, 5 and 8 to download apps through bootloader again.

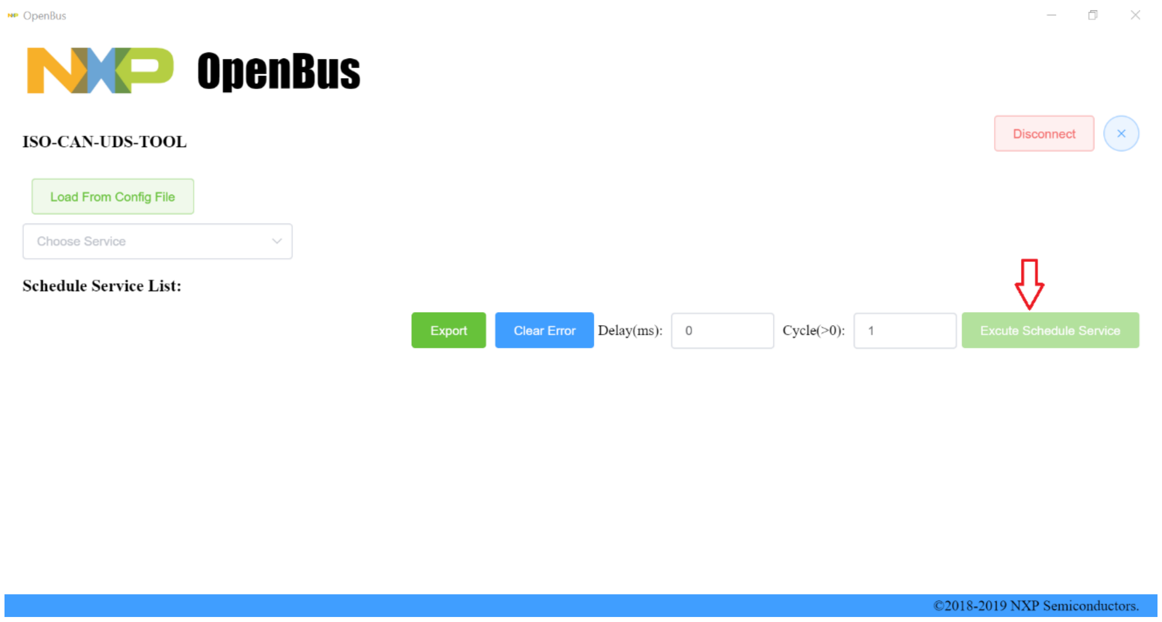


Figure 53. Table properties window