UG10109

EVSE-SIG-BRD1X User Guide

Rev. 1.0 — 18 June 2024 **User guide**

Document information

Information	Content
Keywords	UG10109, EVSE-SIG-BRD1X, LPC5536/LPC55S36, EVSE, EV simulation software, CP, PP
Abstract	This document is a quick and easy guide to get EVSE-SIG-BRD1X up and running and integrate with an NXP host platform evaluation board. The EVSE-SIG-BRD1X is an add-on development board that supports electric vehicle supply equipment (EVSE) or electric vehicle (EV) platform development.

1 Board overview

ISO 15118 is becoming a globally dominant electric vehicle charging standard. The types of electric vehicle (EV) it can support include motor bikes, cars, buses, trucks, boats, and so on. The standard is user-friendly and supports plug and charge technology. When the EV is connected to the charger, the EV automatically identifies itself to recharge its battery. The standard supports strong data security both in transport and application layers using TLS1.2, XML-based signatures, and X.509 certificates. The EV also acts as a battery bank and supply power to the home during the down time of the local energy grid supply. It is done in a mode referred to as bidirectional power transfer (BPT).

ISO 15118 also gives provision for secure and automated billing via an e-mobility operator. The standard helps in load management in the EV for battery-optimized charging. Depending on the dynamic grid supply condition, renegotiation of the charging schedule is achievable to satisfy several EVs charging simultaneously.

NXP enables the host controller support for EV supply equipment (EVSE) from a range of microcontrollers and microprocessors, such as:

- i.MX RT106x crossover MCU
- i.MX 8M Nano applications processor
- i.MX 93 applications processor

Advanced security is provided using add-on modules with secure element (NXP SE050) and near-field communication (NFC) capability for user/EV identification using NFC frontend solutions from NXP.

The EVSE-SIG-BRD1X is an add-on development board that supports EVSE or EV platform development. The main host of the system is on a separate processor development board. For example, NXP i.MX RT106x EVK, i.MX 8M Nano EVK, or S32G-VNP-RDB3. The ISO 15118 stack and communication software run on the host processor. The power-line communication path is via the HomePlug Green PHY (HPGP) transceiver (Lumissil IS32CG5317) included on the EVSE-SIG-BRD1X. The EVSE development platform, including host controller, EVSE-SIG-BRD1X, security and NFC modules, and NXP Kinetis KM3x family of metering microcontroller solutions can form the basis of a full electric vehicle charging station for quick system design and prototyping.

This document is a quick and easy guide to get the EVSE-SIG-BRD1X up and running and integrate with an NXP host platform evaluation board. For EVSE simulation, the host platform board can be a i.MX RT106x EVK board hosting the i.MX RT Crossover MCU, i.MX 8M Nano applications processor, and i.MX 93 applications processor. For EV simulation, the host platform can be S32G-VNP-RDB2/3 or S32K312EVB-Q172 or similar supporting Arduino Uno interface.

Note: *To get implementation details of individual sub-blocks of the board, see the EVSE-SIG-BRD1X User Manual (document UM12013).*

1.1 Block diagram

Figure 1 shows the EVSE-SIG-BRD1X block diagram.

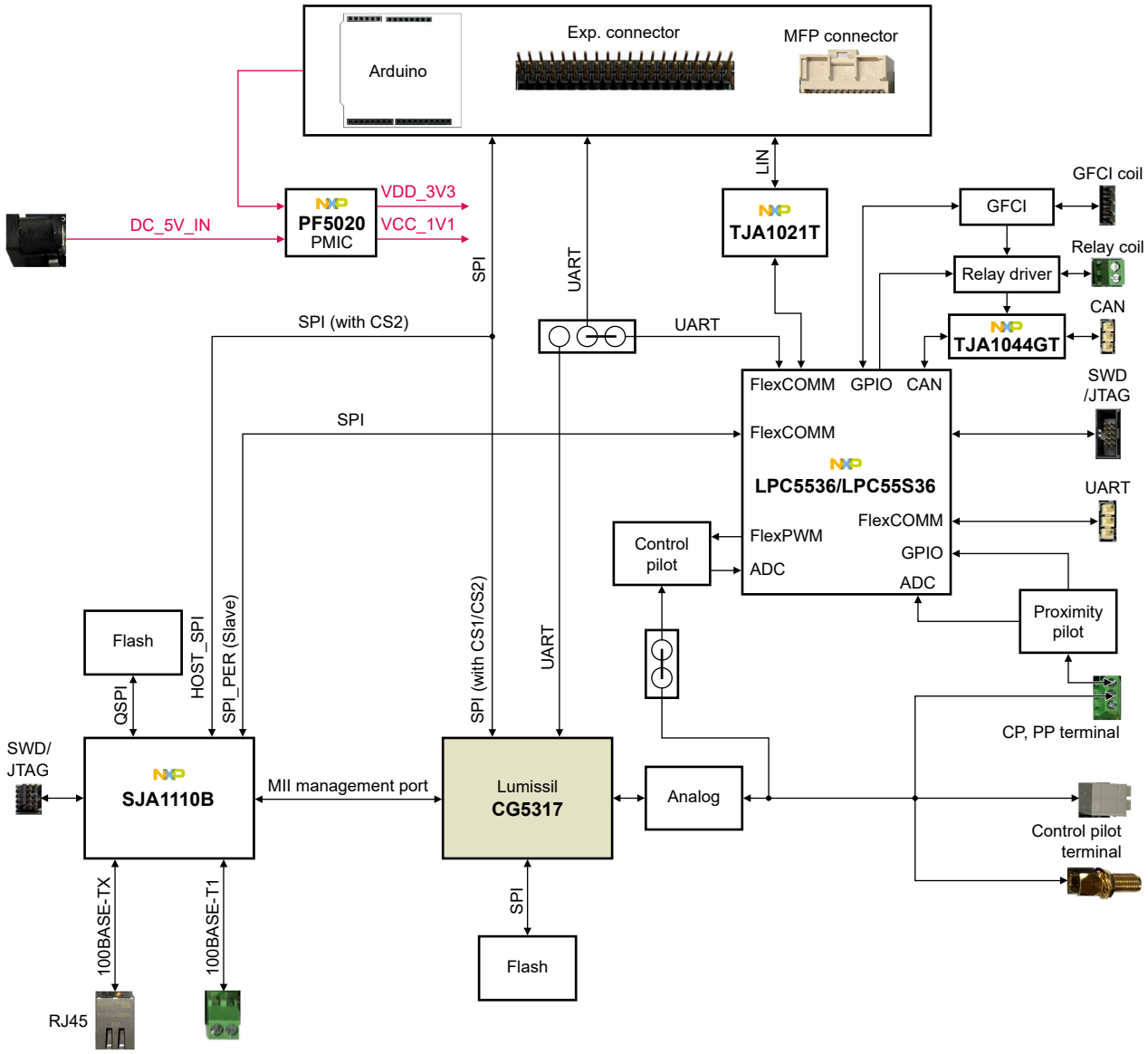


Figure 1. EVSE-SIG-BRD1X hardware block diagram

1.2 Board features

Table 1 lists the board features of the EVSE-SIG-BRD1X.

Table 1. EVSE-SIG-BRD1X features

Board feature	Description
Embedded microcontroller	NXP LPC5536/LPC55S36 MCU, which features a 32-bit Arm Cortex-M33 core, 128 KB SRAM, 256 KB flash, FlexSPI with cache, USB FS, Flexcomm interface, CAN FD, 32-bit counters/timers, SCTimer/PWM, 16-bit 2.0 Msamples/s ADC, comparator, 12-bit DAC, op-amp, FlexPWM timer, QEI, temperature sensor, and CRC
Embedded HPGP	Lumissil CG5317
Embedded Ethernet switch MCU	NXP SJA1110B

Table 1. EVSE-SIG-BRD1X features*...continued*

Board feature	Description
Host connectors • Arduino Uno • ECP CN/GPIO header • Multi-function port (MFP) connector	• Power: +5 V, +3.3 V • One SPI port with two chip selects • One UART port • GPIOs • LIN (MFP only)
Ethernet host interface	• One100BASE-TX • One 100BASE in place of 100BAST-T1 interfaces
CAN interface	One NXP TJA1044GT CAN PHY interface
LIN interface	One NXP TJA1021T/20/C LIN PHY interface
Debug interface	• Auxiliary UART interface from LPC5536/LPC55S36 • SWD debug port of LPC5536/LPC55S36 for development
Control pilot	J1772 (IEC 61851) PWM, ISO 15118-2/20 EVSE, and EV support
Proximity pilot	J1772 support
GFCI	GFCI detection and relay asynchronous triggering
Relay driver	Drive up to two DC coil relays at 12 V, 140 mA
Power	• Primary power supply options: – 5 V external power through DC power supply jack (J1) – Power from the host connector (Arduino/EXP CN/MFP) • On board, +5 V to +12 V boost converter • On board, +12 V to -12 V charge pump inverter
PCB	6.4 inch x 3 inch, 6-layer
Orderable part number	EVSE-SIG-BRD1X

1.3 Board pictures

Figure 2 shows the top-side view of the EVSE-SIG-BRD1X.

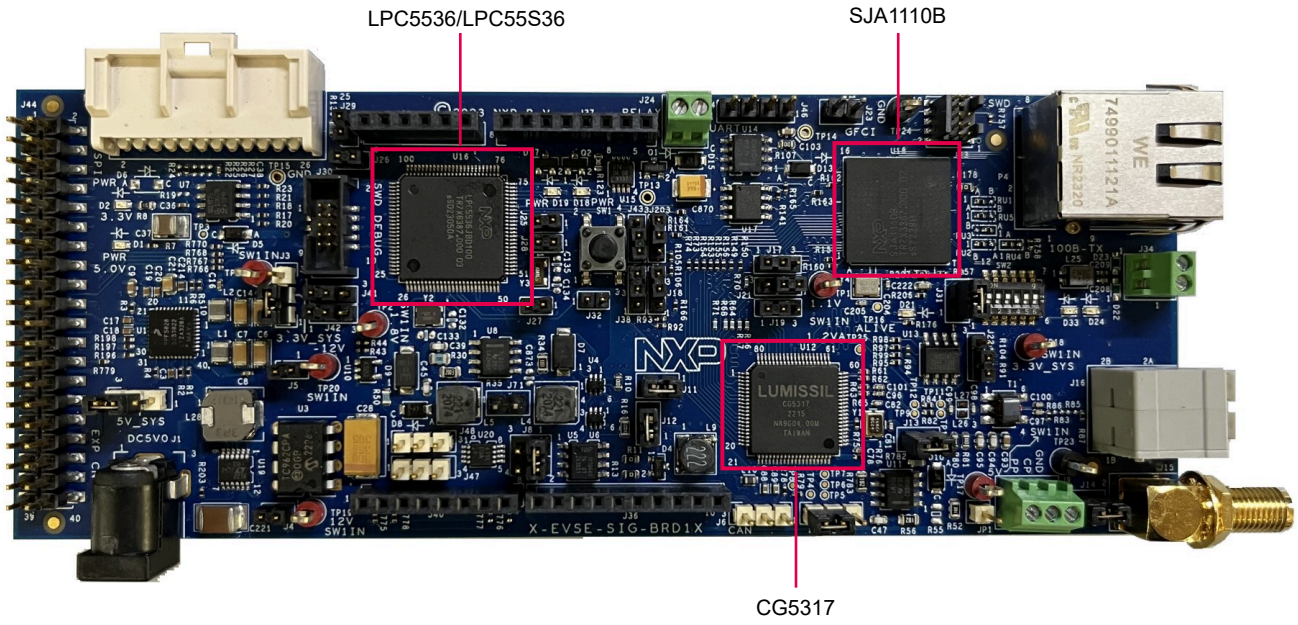


Figure 2. EVSE-SIG-BRD1X top-side view

Figure 3 shows the bottom-side view of the EVSE-SIG-BRD1X.

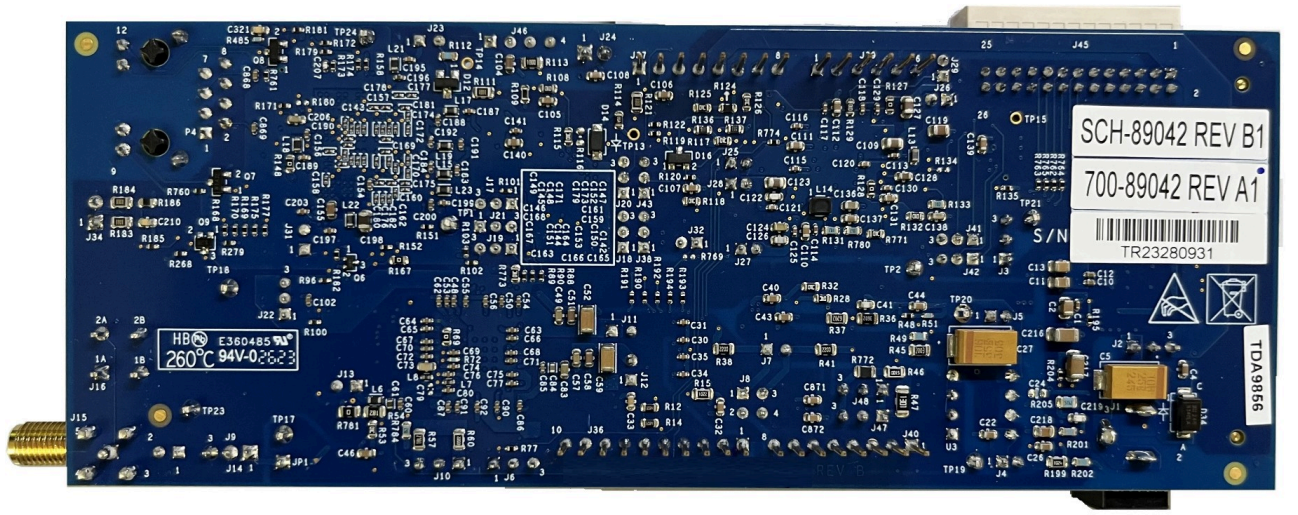


Figure 3. EVSE-SIG-BRD1X bottom-side view

1.4 Connectors

Figure 4 shows the EVSE-SIG-BRD1X connectors.

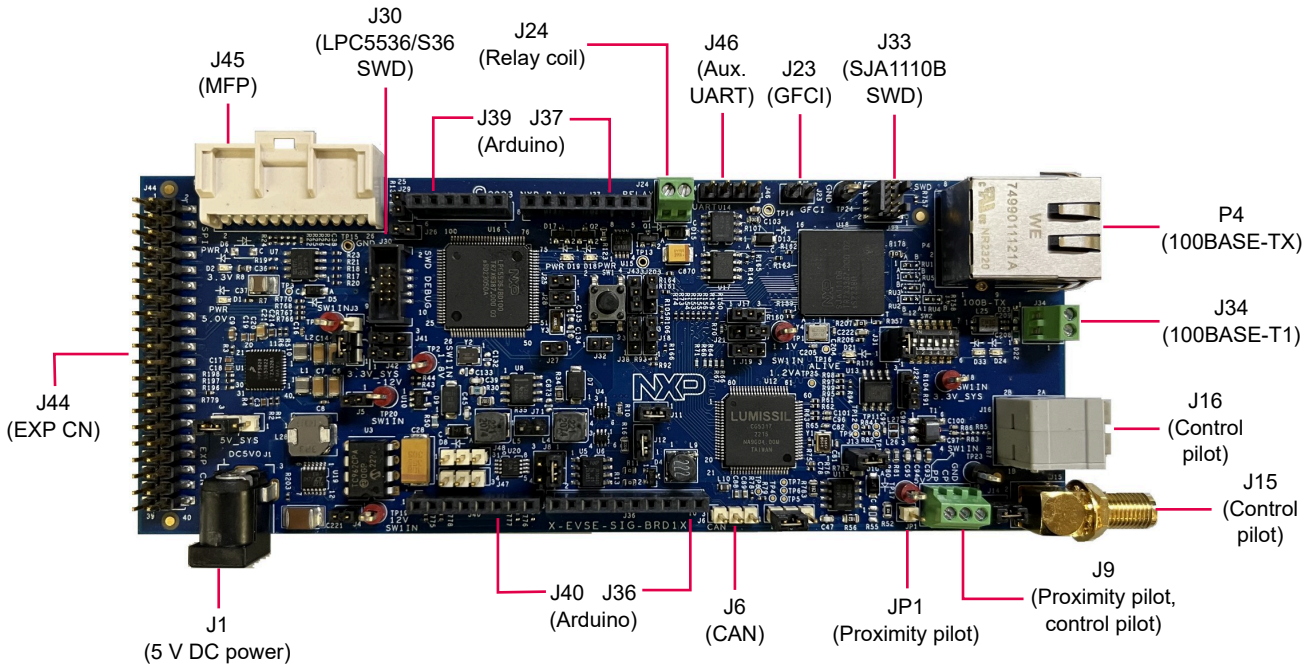


Figure 4. EVSE-SIG-BRD1X connectors

Table 2 describes the EVSE-SIG-BRD1X connectors.

Table 2. EVSE-SIG-BRD1X connectors

Part identifier	Connector type/default	Description
J1	DC power jack; open	External +5 V power supply jack
J6	1x3-pin header	HS CAN connector
J9	3-position wire-to-board connector	Control pilot/proximity pilot connector
J15	SMA receptacle	Control pilot (test) connector
J16	2x2-position receptacle	Control pilot connector
J23	1x2-pin header	Secondary GFCI coil connector
J24	2-position wire-to-board connector	Relay coil connector
J30	2x5-pin header	LPC5536/LPC55S36 SWD debug connector
J33	9-pin (10-position) header	SJA1110B SWD debug connector
J34	2-position wire-to-board connector	100BASE-T1 Ethernet connector
J36	1x10-position receptacle	Arduino Uno connector that provides power to the GPIO signals
J37	1x8-position receptacle	Arduino Uno connector that provides power to the board
J39	1x6-position receptacle	Arduino Uno connector that provides power to the GPIO signals
J40	1x8-position receptacle	Arduino Uno connector that provides power to the GPIO signals
J44	2x20-pin header	Expansion connector that provides power to the board, GPIO signals

Table 2. EVSE-SIG-BRD1X connectors*...continued*

Part identifier	Connector type/default	Description
J45	2x13-position receptacle	MFP connector that provides power to the board, GPIO signals, LIN signals
J46	1x4-pin header; open	Auxiliary UART connector
JP1	1-pin header	Proximity pilot connector
P4	RJ45 jack	100BASE-TX Ethernet connector

1.5 Jumpers

Table 3 shows the EVSE-SIG-BRD1X jumpers.

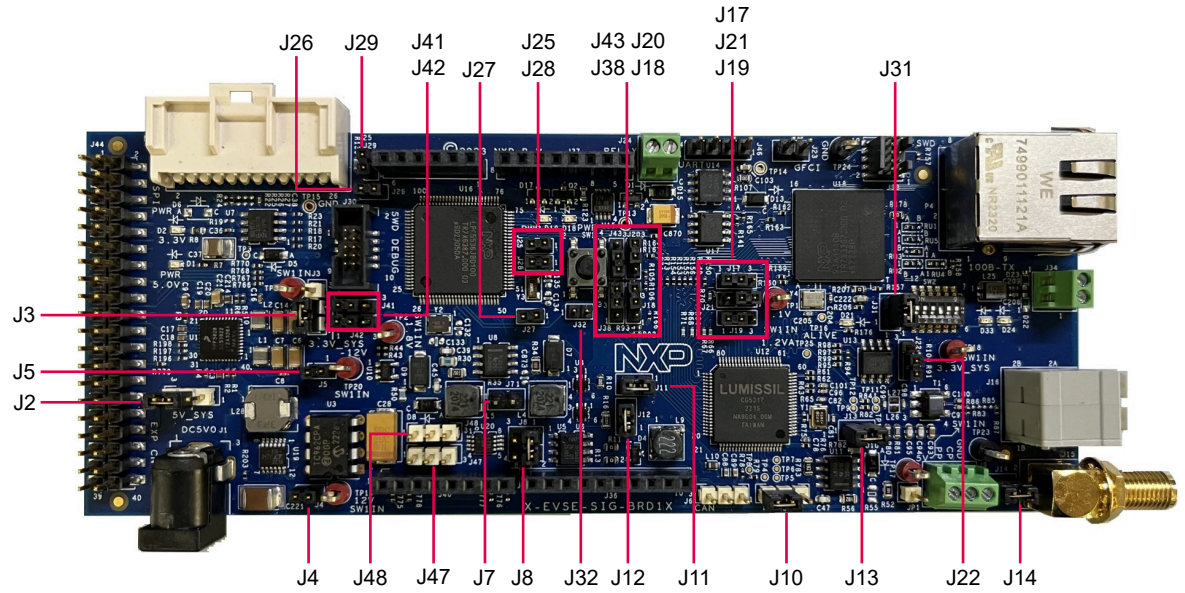


Figure 5. EVSE-SIG-BRD1X onboard jumpers

Table 3 describes the EVSE-SIG-BRD1X jumpers.

Table 3. EVSE-SIG-BRD1X jumpers

Part identifier	Jumper type/default	Description
J2	1x3-pin header	5V_SYS power source selection jumper: • Pins 1-2 shorted: 5V_SYS supply is produced from DC_5 V_IN supply. • Pins 2-3 shorted (default setting): 5V_SYS supply is produced from 5V_ARD_EXP_CN supply.
J3	1x3-pin header	3.3V_SYS power source selection jumper: • Pins 1-2 shorted: 3.3V_SYS supply is produced from VDD_3V3 supply. • Pins 2-3 shorted (default setting): 3.3V_SYS supply is produced from 3V3_ARD_EXP_CN supply.
J4	1x2-pin header	12V0_ISO supply enable jumper: • Open: 12V0_ISO supply is OFF. • Shorted (default setting): 12V0_ISO supply is produced from a 12V0 supply.

Table 3. EVSE-SIG-BRD1X jumpers*...continued*

Part identifier	Jumper type/default	Description
J5	1x2-pin header	-12V0_ISO supply enable jumper: • Open: -12V0_ISO supply is OFF. • Shorted (default setting): -12V0_ISO supply is produced from -12V0 supply.
J7	1x2-pin header	EVSE/EV PWM loopback enable jumper: • Open (default setting): EVSE/EV PWM loopback is disabled. • Shorted: EVSE/EV PWM loopback is enabled.
J8	2x2-pin header	Control pilot selection jumper: • Pins 1-2 shorted (default setting): EVSE control pilot is selected for PWM generation and detection. • Pins 3-4 shorted: EV control pilot is selected for PWM generation and detection.
J10	1x3-pin header	Proximity pilot board test points • Pins 1-2 shorted (default setting): Proximity pilot is used for EVSE simulation. • Pins 2-3 shorted: Proximity pilot is used for EV simulation.
J11	1x2-pin header	3V3_CG5317 supply enable jumper: • Open: 3V3_CG5317 supply is OFF. • Shorted (default setting): 3V3_CG5317 supply is produced from 3.3V_SYS supply.
J12	1x2-pin header	VCORE supply enable jumper: • Open: VCORE supply is OFF. • Shorted (default setting): VCORE supply is produced from 3V3_CG5317 supply.
J13	1x2-pin header	3.3VA supply enable jumper: • Open: 3.3VA supply is OFF. • Shorted (default setting): 3.3VA supply is produced from 3.3V_SYS supply.
J14	1x2-pin header	EVSE/EV control pilot I/O control jumper: • Open: EVSE/EV control pilot is disconnected from J15/J16. • Shorted (default setting): EVSE/EV control pilot is connected to J15/J16.
J17	1x3-pin header	Boot strap pin headers for CG5317 HPGP • Pins 1-2 connected
J18		Boot strap pin headers for CG5317 HPGP • Pins 2-3 connected
J19		Boot strap pin headers for CG5317 HPGP • Pins 1-2 connected
J20		Boot strap pin headers for CG5317 HPGP • Pins 2-3 connected
J21		Boot strap pin headers for CG5317 HPGP • Pins 1-2 connected
J22		Boot strap pin headers for CG5317 HPGP

Table 3. EVSE-SIG-BRD1X jumpers*...continued*

Part identifier	Jumper type/default	Description
		• Pins 1-2 connected
J25	1x2-pin header	MCU_VDD supply enable jumper: • Open: MCU_VDD supply is OFF. • Shorted (default setting): MCU_VDD supply is produced from 3.3V_SYS supply.
J26	1x2-pin header	MCU_VDDA supply enable jumper: • Open: MCU_VDDA supply is OFF. • Shorted (default setting): MCU_VDDA supply is produced from 3.3V_SYS supply.
J27	1x2-pin header	MCU_MAIN supply enable jumper: • Open: MCU_MAIN supply is OFF. • Shorted (default setting): MCU_MAIN supply is produced from 3.3 V_SYS supply.
J28	1x2-pin header	MCU_VBAT supply enable jumper: • Open: MCU_VBAT supply is OFF. • Shorted (default setting): MCU_VBAT supply is produced from 3.3 V_SYS supply.
J29	1x2-pin header	LPC5536/LPC55S36 MCU boot mode selection jumper: • Open: LPC5536/LPC55S36 MCU boots in In-System Programming (ISP) mode. • Shorted (default setting): LPC5536 MCU boots in Normal mode
J31	1x2-pin header	VCC_3V3_S supply enable jumper: • Open: VCC_3V3_S supply is OFF. • Shorted (default setting): VCC_3V3_S supply is produced from 3.3V_SYS supply.
J32	1x2-pin header	SJA1110B SPI host connection enable jumper: • Open (default setting): The SJA1110B SPI interface cannot connect to a host controller board. • Shorted: SJA1110B SPI interface (master) can connect to a host controller board (slave).
J38	1x3-pin header	HPGP (CG5317) SPI master selection jumper: • Pins 1-2 shorted (default setting): Host controller SPI chip select 1 is connected. • Pins 2-3 shorted: Host controller SPI chip select 2 is connected.
J41 J42	1x3-pin header	Arduino socket connector J40 UART port control jumpers: • Pins 1-2-3 open: J40 UART port is connected to the expansion connector J44 UART port. • Pins 1-2 shorted (default setting): J40 UART port is connected to the LPC5536 MCU UART port. • Pins 2-3 shorted: J40 UART port is connected to the HPGP (CG5317 PHY) UART port.
J43	1x3-pin header	Host controller SPI interrupt source selection jumper: • Pins 1-2 shorted (default setting): HPGP (CG5317) SPI interface is selected as the interrupt source.

Table 3. EVSE-SIG-BRD1X jumpers*...continued*

Part identifier	Jumper type/default	Description
		• Pins 2-3 shorted: SJA1110B switch is selected as the interrupt source.
J47 J48	1x3-pin header	EV charging ventilation option selection jumper: • Pins 1-2 shorted (default setting): 1.3 kΩ resistance (R47) is ON; the vehicle can be charged in an unventilated area. • Pins 2-3 shorted: 530 Ω resistance (R46 + R772) is ON; the vehicle can be charged only in a ventilated area. ***Note:*** *Both J47 and J48 serve the same purpose. J47 is used for MCU-controlled switching whereas J48 is used for manual switching. Only one of them can be used at a time. By default, J47 is used.*

1.6 Push button and DIP switch

The EVSE-SIG-BRD1X has one push button SW1 and one dual inline package (DIP) switch SW2, as shown in Figure 6.

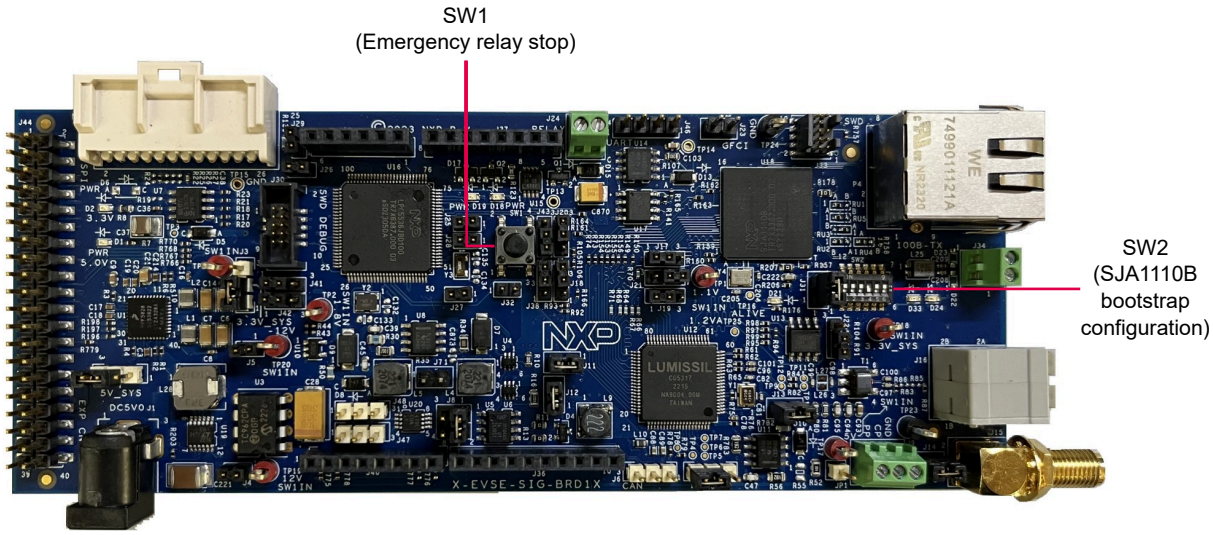


Figure 6. Push button and DIP switch

Table 4 describes the EVSE-SIG-BRD1X push button.

Table 4. EVSE-SIG-BRD1X push button

Part identifier	Supported function	Description
SW1	Emergency relay stop button	This push button can be used to turn OFF the relay during an emergency. Usually, the LPC5536 / LPC55S36 MCU is used to turn ON / turn OFF the relay.

SW2 is a 6-pin DIP switch for manually controlling the power-on bootstrap functions of the SJA1110B Ethernet switch on the EVSE-SIG-BRD1X.

Each pin of the DIP switch has the following two positions:

- OFF position (pin has value 0)
- ON position (pin has value 1)

A DIP switch pin can be moved manually from OFF position to ON position and vice versa.

Table 5 describes the DIP switch SW2.

Table 5. EVSE-SIG-BRD1X DIP switch

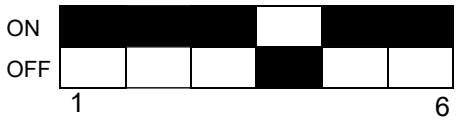
Part identifier	Switch name	Description
SW2	Boot strap switch for SJA1110	 6-pin DIP switch

Table 6 describes SW2 settings / SJA1110B bootstrap configuration.

Table 6. SW2 settings / SJA1110B bootstrap configuration

Pin pair	Description
SW2[1]	1: Always ON position (default setting)
SW2[2:3]	BOOT_OPTION[0:1]: • 00: Serial Download mode. An image is downloaded at Linux boot time. • 01: Boot from EEPROM (reserved) • 10: Boot from SPI flash • 11: Boot from QSPI flash (default setting)
SW2[4]	PHY_M_S5: • 0: PHY slave port (default setting) • 1: PHY master port
SW2[5]	PHY_AUTO_POL_DET: • 0: If polarity is wrong, link training is blocked • 1: Fully automated polarity detection and correction for 100BASE-T1 PHY port 5 (default setting)
SW2[6]	PHY_AUTO_MODE: Automatic mode select: • 0: Managed mode • 1: Automatic mode. The 100BASE-T1 PHY starts link training automatically (default setting).

1.7 LEDs

The EVSE-SIG-BRD1X has light-emitting diodes (LEDs) to monitor system functions. The information collected from LEDs can be used for debugging purposes.

Figure 7 shows the EVSE-SIG-BRD1X LEDs.

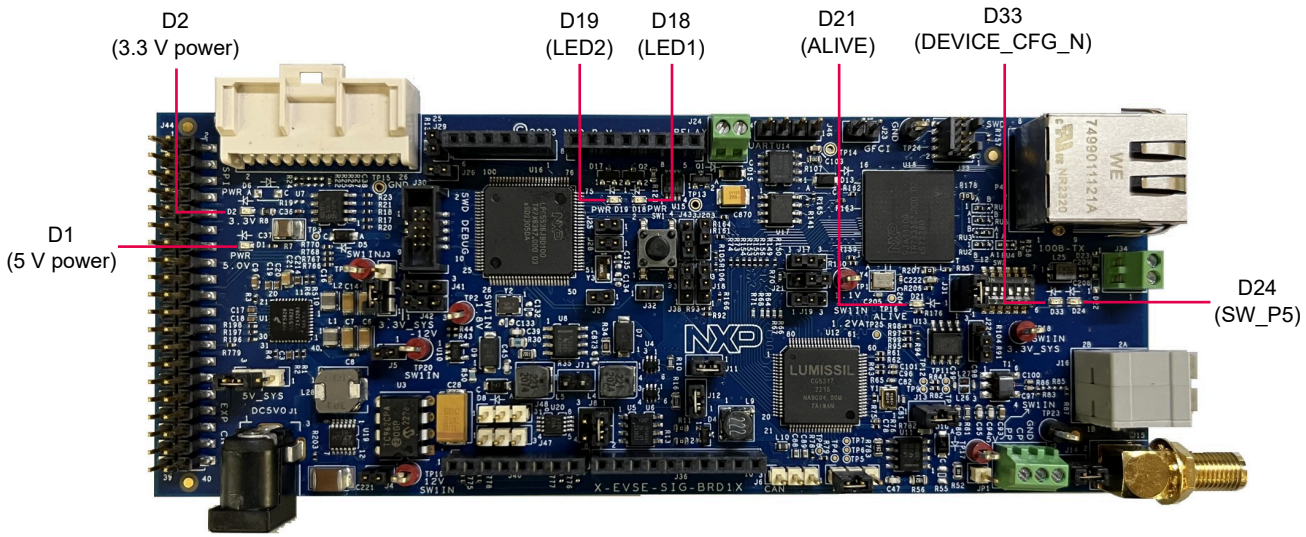


Figure 7. LEDs

Table 7 describes the EVSE-SIG-BRD1X LEDs.

Table 7. EVSE-SIG-BRD1X LEDs

Part identifier	PCB label	LED color	Description (when LED is ON)
D1	PWR 5.0 V	Green	5V_SYS supply is available
D2	PWR 3.3 V	Green	3.3V_SYS supply is available
D18	PWR	Green	User application LED 1
D19	PWR	Green	User application LED 2
D21	ALIVE	Green	SJA1110B is up and running
D24		Green	Link activity is in progress for SJA1110B switch 100BASET1 port 5.
D33		Green	SJA1110B switch subsystem configuration is complete

2 Featured interfaces

This section contains the following subsections:

2.1 Power supply

The EVSE-SIG-BRD1X draws power from the host EVK connectors, for example, Arduino, EXP-CN, or MFP. To drive external relays (140 mA or above generated at 12 V), the board has a DC power jack for supplying 5 V external power.

Table 8 shows the EVSE-SIG-BRD1X power supply sources.

Table 8. Power supply sources

Power source	Power supply rail	Description
Through J37 pin 5	5V_ARD_EXP_CN	The power supply is received from a host board with the Arduino, EXP CN, or MFP (S32G-VNP-RDB2)
Through J44 pins 2 and 4		
Through J45 pin 1		
Through the DC power jack, J1	DC_5V_IN	The power supply is received through an external 5 V DC power source
Through J2 pin 2	5V_SYS	Board 5 V power; it can be either selected from 5V_ARD_EXP_CN or DC_5V_IN or supplied from an external power source
Through J37 pin 4	3V3_ARD_EXP_CN	The power supply is received from a host board with the Arduino, EXP CN, or MFP (S32G-VNP-RDB2)
Through J44 pins 1 and 17		
Through J45 pin 3		
Through J3 pin 2	3.3V_SYS	Board 3.3 V power; it can be either selected from 3V3_ARD_EXP_CN or VDD_3V3 or supplied from an external power source
U1	VCC_1V1	1.1 V power supply for the SJA1110B switch core
U1	VDD_3V3	3.3 V output from PMIC
U19	12V0	Board 12 V power that powers the control pilot PWM op-amp and the relay MOSFET
U3	-12V0	Board 12 V power that powers the control pilot PWM op-amp

2.2 Proximity pilot

Table 9 shows the proximity pilot signal with the connector positions available on the board.

Table 9. EVSE-SIG-BRD1X jumper settings for proximity pilot

Connector	Connector type	Pin	Signal name	Description
J9	Terminal block	1	PROX_PILOT (PP)	Proximity pilot signal on terminal connector
JP1	Test point header	1	PROX_PILOT (PP)	Proximity pilot test point

2.3 Control pilot

Table 10 shows the control pilot signal with the connector positions available on the board.

Table 10. EVSE-SIG-BRD1X connector settings for control pilot

Connector	Connector type	Pin	Signal name	Description
J9	Terminal block	2	CP	J1772 control pilot PWM
J15	SMA	1	CP_ANA	ISO 15118 control pilot
J16	Terminal block	2A, 2B	CP_ANA	ISO 15118 control pilot

Table 11 summarizes the bootstrap jumpers of HPGP CG5317.

Table 11. CG5317 boot strap pins

Jumper	Bootstrap function description
J21, J17, J19	MII_PHY_ADD[2-0] of CG5317 Ethernet PHY: • J21, J19, J17 are configured as 2:0, where the value is '1' for position 1, 2 and '0' for position 3. Therefore, the default value is 011.
J18	SPI_CLK_MODE: • Pins 1 and 2 shorted = Value '1' = SPI MODE 1 • Pins 2 and 3 shorted = Value '0' = SPI MODE 3 (default)
J20	BOOT_SRC: • Pins 1 and 2 shorted = Value '1' = HOST • Pins 2 and 3 shorted = Value '0' = AUTO (default)
J22	UART_DISABLE: • Pins 1 and 2 shorted = Value '1' = DISABLE (default) • Pins 2 and 3 shorted = Value '0' = ENABLE

Note: *Ensure that the power isolation jumpers J11, J12, and J13 are always short.*

2.4 GFCI circuit

Table 12 shows the GFCI coil input signal available on the board with the connector positions.

Table 12. EVSE-SIG-BRD1X connector settings for GFCI circuit

Connector	Connector type	Pin	Signal name	Description
J23	Header	1, 2	GFCI coil terminal pair	Connect the external GFCI coil here

2.5 Relay driver circuit

Table 13 shows the relay driver DC coil output signal available on the board with the connector positions.

Table 13. EVSE-SIG-BRD1X connector settings for relay driver circuit

Connector	Connector type	Pin	Signal name	Description
J24	Terminal block	1, 2	Relay coil connector	Connect the external relay here

2.6 Host connectors

EVSE-SIG-BRD1X supports connection for several different host processor development boards, via a SPI connection to the HPGP and a UART to the onboard LPC5536/LPC55S36 MCU.

The supported connections are listed below:

- Arduino UNO R3 connectors, which fetch power from the host board and support SPI and UART connections. The Arduino interface is used to connect the EVSE-SIG-BRD1X to a development board for the i.MX RT crossover MCU (MIMXRT1060-EVK) or an S32K3 automotive MCU (S32K3X4EVB-T172).
- i.MX EXP-CN connector, which is an alternative to Arduino UNO and provides easy connection to i.MX evaluation kit (EVK) boards, for example 8MNANOD4-EVK and i.MX93EVK.
- MFP, which provides an easy connection to the S32G-VNP-RDB2 or S32G-VNP-RDB3 development board.

Table 14, Table 15, Table 16, and Table 17 show the pinouts of the Arduino connectors J37, J39, J36, and J40, respectively.

Table 14. Arduino connector J37 pinout

Pin numbers	Signal name	Type	Description
4	3V3_ARD_EXP_CN	Power	+3.3 V
5	5V_ARD_EXP_CN	Power	+5 V
6, 7	Ground		Ground
1, 2, 3, 8			Not connected

Table 15. Arduino connector J39 pinout

Pin numbers	Signal name	Type	Description
1	HPGP_GP_IRQ	O	HPGP general-purpose interrupt
3	HOST_SPI_IRQ	O	HPGP_SPI_IRQ || SW_INT_N (selection through J43): • HPGP_SPI_IRQ: HPGP SPI interrupt • SW_INT_N: SJA1110B switch interrupt
4	HPGP_RESET	I	HPGP reset signal
2, 5, 6			Not connected

Table 16. Arduino connector J36 pinout

Pin numbers	Signal name	Type	Description
2	HOST_SPI_CS1	I	Host SPI master chip select option 1
3	HOST_SPI_CS2	I	Host SPI master chip select option 2
4	HOST_SPI_MOSI	I	Host SPI master output slave input signal
5	HOST_SPI_MISO	O	Host SPI master input slave output signal
6	HOST_SPI_CLK	I	Host SPI master clock
7	Ground		Ground
1, 8, 9, 10			Not connected

Table 17. Arduino connector J40 pinout

Pin numbers	Signal name	Type	Description
1	HOST_UART_TXD	O	Host UART transmit
2	HOST_UART_RXD	I	Host UART receive
3, 4, 5, 6, 7, 8			Not connected

Table 18 shows the pinout of the EXP CN connector J44.

Table 18. EXP CN connector J44 pinout

Pin numbers	Signal name	Type	Description
1, 17	3V3_SRD_EXP_CN	Power	+3.3 V
2, 4	5V_ARD_EXP_CN	Power	+5 V
7	HOST_SPI_IRQ	O	HPGP_SPI_IRQ || SW_INT_N (selection through J43): • HPGP_SPI_IRQ: HPGP SPI interrupt • SW_INT_N: SJA1110B switch interrupt
8	HOST_UART_RXD	I	Host UART receive
10	HOST_UART_TXD	O	Host UART transmit
11	HPGP_GP_IRQ	O	HPGP General-Purpose Interrupt
12	HPGP_RESET	I	HPGP Reset signal
16	HPGP_SPI_CS2	I	Host SPI master chip select option 2
19	HOST_SPI_MOSI	I	Host SPI master output slave input signal
21	HOST_SPI_MISO	O	Host SPI master input slave output signal
23	HOST_SPI_CLK	I	Host SPI master Clock
24	HPGP_SPI_CS1	I	Host SPI master chip select option 1
6, 9, 14, 20, 25, 30, 34, 39	Ground		Ground
3, 5, 13, 15, 18, 22, 26, 27, 28, 29, 31, 32, 33, 35, 36, 37, 38, 40			Not connected

Table 19 shows the pinout of the MFP connector J45.

Table 19. MFP connector J45 pinout

Pin numbers	Signal name	Type	Description
1	5V_ARD_EXP_CN	Power	+5 V
3	3V3_ARD_EXP_CN	Power	+3.3 V
9	LIN1	I/O	LIN master 1
10	LIN0	I/O	LIN master 0
11	LIN3	I/O	LIN master 3
12	LIN2	I/O	LIN master 2
13	HPGP_GP_IRQ	O	HPGP general-purpose interrupt
14	HOST_SPI_IRQ	O	HPGP_SPI_IRQ || SW_INT_N (selection through J43): • HPGP_SPI_IRQ: HPGP SPI interrupt • SW_INT_N: SJA1110B switch interrupt
15	HOST_SPI_MISO	O	Host SPI master input slave output signal
16	HOST_SPI_MOSI	I	Host SPI master output slave input signal

Table 19. MFP connector J45 pinout*...continued*

Pin numbers	Signal name	Type	Description
17	HOST_SPI_CS1	I	Host SPI master chip select option 1
18	HOST_SPI_CLK	I	Host SPI master clock
5, 6, 7, 19, 20, 21	Ground		Ground
2, 4, 8, 22, 23, 24, 25, 26			Not connected

2.7 CAN PHY

The board supports a CAN PHY (NXP TJA1044GT) on board to communicate with other devices in the EV/ automotive use case.

Table 20. CAN header connector J6 pinout

Pin	Signal name	Type	Description
1	Ground		
2	CANH	IO	CAN differential high signal
3	CANL	IO	CAN differential low signal

2.8 LIN PHY

The MFP supports local interconnect network (LIN) interfaces through LIN PHY TJA1021T at connector J45 pins 9, 10, 11, and 12.

2.9 Auxiliary / debug UART port

An additional UART port is provided to integrate an external meter board for communication. This option is applicable when the host controller board does not communicate directly with the meter board due to connectivity limitation.

The port can be used to print serial debug logs of LPC5536. Connect an external PC UART TTL serial port adapter to J46 pins 2, 3, and 4 as shown in Table 21.

Table 21. Auxiliary header connector J46 pinout

Pin	Signal name	Type	Description
1	MCU_VDD	Power	+3.3 V
2	M_UART_TXD	O	UART transmit
3	M_UART_RXD	I	UART receive
4	Ground		Ground

2.10 LPC5536/LPC55S36 MCU

EVSE-SIG-BRD1X hosts an LPC5536/LPC55S36 MCU controller to support the required local controller functions of the board. Therefore, this MCU acts as a utility controller for the EVSE system.

2.10.1 LPC5536/LPC55S36 SWD debug

EVSE-SIG-BRD1X provides a single wire debug (SWD) port through connector J30 for debugging the LPC5536/LPC55S36 MCU.

User can use an MCU-Link Debug Probe, MCU-Link Pro Debug Probe, or PE micro to program and debug this MCU.

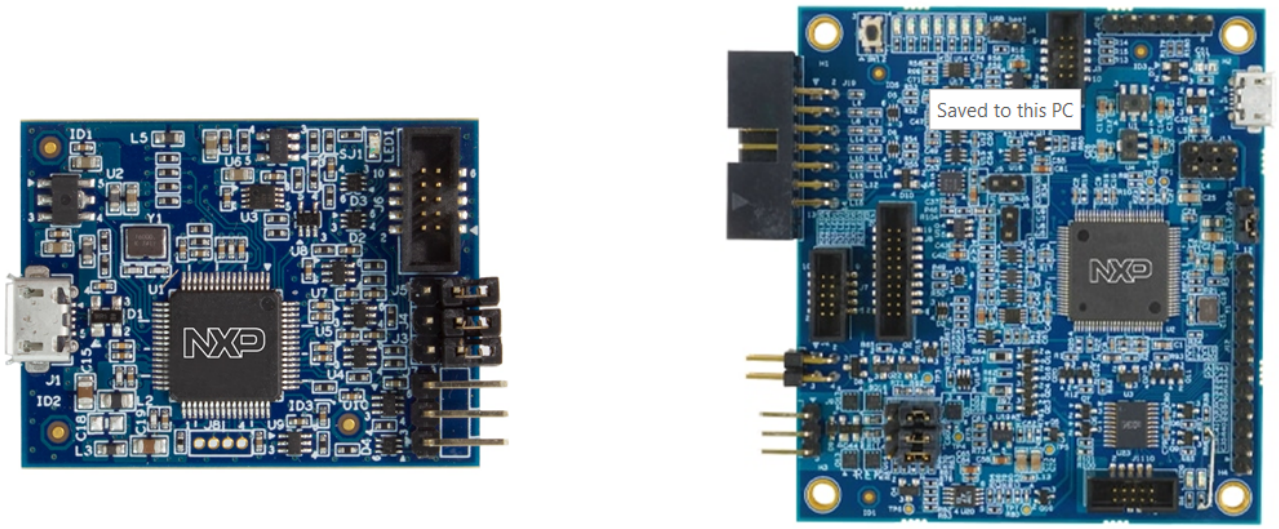


Figure 8. MCU-Link Debug Probe and MCU-Link Pro Debug Probe

2.10.2 LPC5536/LPC55S36 ISP programming

EVSE-SIG-BRD1X uses In-System Programming (ISP) through the UART interface to program LPC5536/LPC55S36. UART peripheral implements auto-baud detection. To set up LPC5536/LPC55S36 for ISP programming, the ISP mode selection jumper J29 must be changed from the default shorting state to open.

Table 22 shows the setting of J29 for boot mode selection.

Table 22. EVSE-SIG-BRD1X LPC5536/LPC55S36 boot mode selection

Jumper 29 state	Boot mode
Closed (default)	Internal flash boot
Open	ISP boot

3 EVSE-SIG-BRD1X EVSE configuration

To simulate an EVSE system with the EVSE-SIG-BRD1X, one of the compatible host platforms can be used. The compatible interfaces suggested for the EVSE host are as follows:

- A host board with an Arduino Uno interface; for example, MIMXRT1064-EVK, MIMXRT1060-EVKB
- A host board with EXP CN/GPIO connector interface; for example, 8MNANOD4-EVK, i.MX93EVK

3.1 Host controller i.MX RT1060-EVKB

Figure 9 shows a connection diagram of the i.MX RT1060-EVKB host board with EVSE-SIG-BRD1X.

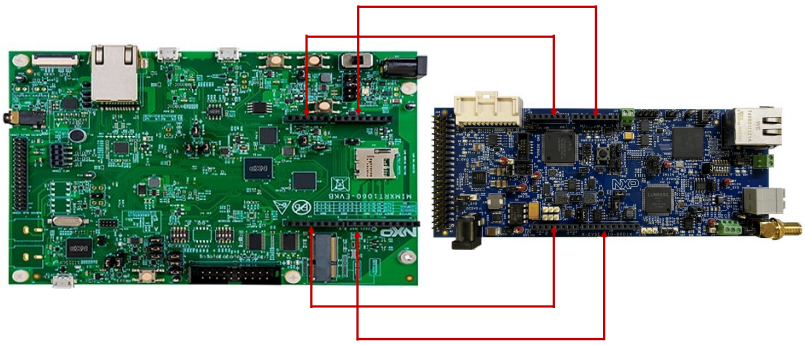


Figure 9. Connection between i.MX RT1060-EVKB and EVSE-SIG-BRD1X

Figure 10 shows i.MX RT1060-EVKB and EVSE-SIG-BRD1X after the interconnection.

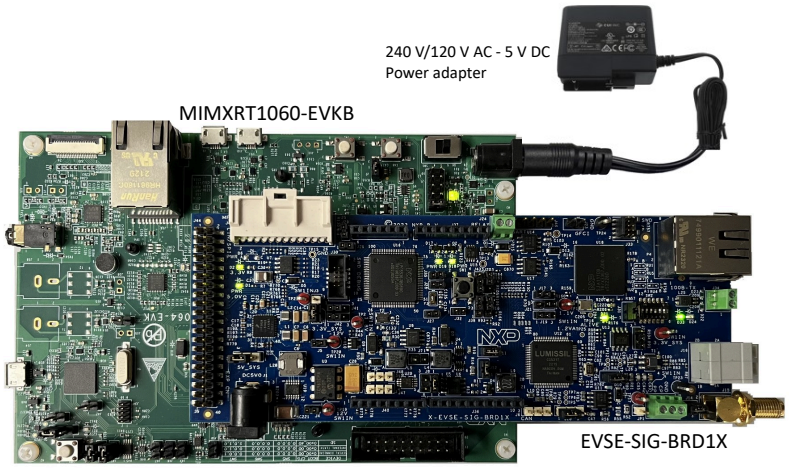


Figure 10. i.MX R106x-EVK and EVSE-SIG-BRD1X connected

Also, peripheral boards for NFC, secure element, WiFi adapter, and TFT display can also be connected as described in EasyEVSE platform documents.

3.2 Host controller i.MX 8M Nano-EVK

Figure 11 shows a connection diagram of the i.MX 8M Nano-EVK host board with EVSE-SIG-BRD1X.

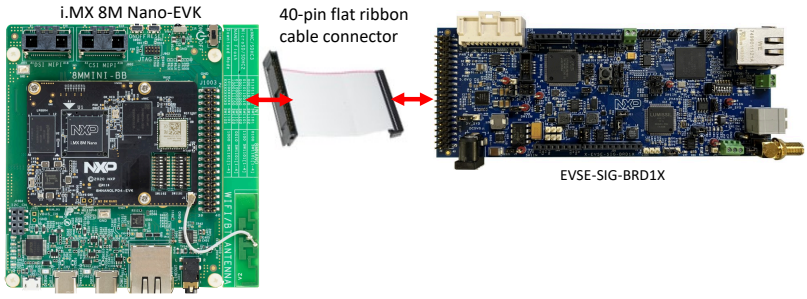


Figure 11. Connection between i.MX 8M Nano-EVK and EVSE-SIG-BRD1X

3.3 EVSE configuration

3.3.1 Proximity pilot sense

The proximity pilot terminal is available at connector J9 pin 1, which can be connected to the proximity pilot wire of the charging cable.

Alternatively, the JP1 jumper point allows the access to the signal.

Depending on when the board is used in the EVSE configuration or EV configuration, there is a different configuration for proximity pilot on the board. The jumper header J10 is shorted through pins 1 and 2 by default for the EVSE configuration.

To set up the proximity pilot for EV configuration, perform the following steps:

1. Remove the resistor 0 Ω R781 on the bottom side of the board.
2. Solder it to the empty resistor R55 footprint on the top side of the board.
3. Also, move jumper header J10 short to pins 2 and 3 from the default position 1 and 2.

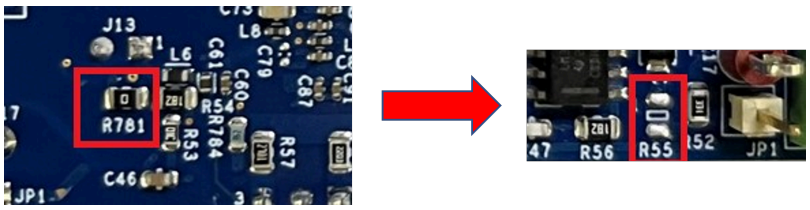
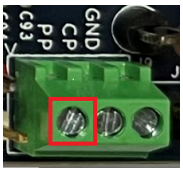


Figure 12. Set up the proximity pilot for EV configuration

Table 23. Connectors/jumpers for proximity pilot in EVSE/EV setup

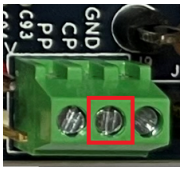
Terminal connector/Jumper setup	Setting	Description
J9	Pin 1	
J10	• J10 pins 1 and 2 short (default): EVSE mode • J10 pins 2 and 3 short: EV mode	

3.3.2 Control pilot

In the EVSE-SIG-BRD1X, the jumper J8 pins 1 and 2 must be short to route the PWM signal to the terminal connector J9 pin 2.

The HPGP CG5317 on board does the ISO 15118 signal and data communication. To combine the J1772 PWM to the ISO 15118 signal, the jumper J14 pins 1 and 2 must be short. It results in a combined J1772 and ISO 15118 signal output at the terminal connector J9 pin 2.

Table 24. Connectors/jumpers for control pilot in EVSE setup

Terminal connector/Jumper setup	Setting	Description
J9	Pin 2	 J9 pin 2 is the control pilot signal in the terminal
J8	Pins 1 and 2 short	 Short pins 1 and 2 for EVSE mode
J14	Pins 1 and 2 short	 Short pins 1 and 2 to combine PWM signal to ISO 15118 signal

3.3.3 GFCI

GFCI coil can be connected to EVSE-SIG-BRD1X at header J23 between pins 1 and 2.

3.3.4 Relay connection

One or two external relays with the DC coils operating at 12 V can be driven using the relay driver connector J24 pins 1 and 2.

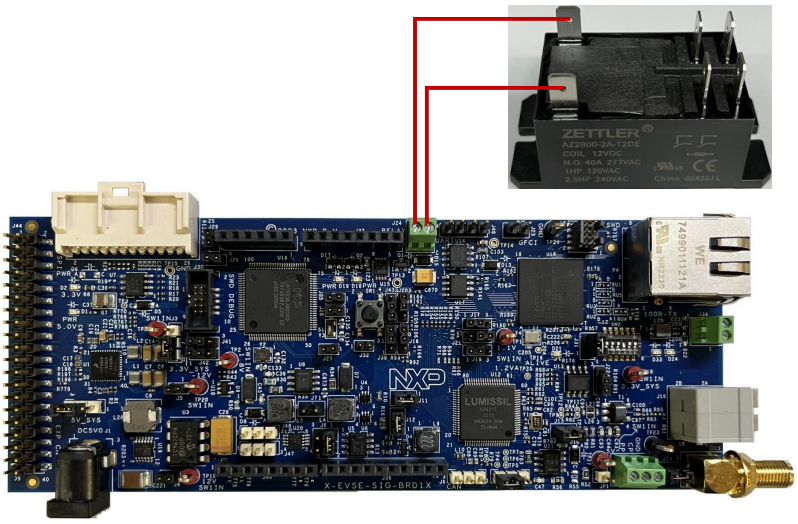


Figure 13. External relay connection to EVSE-SIG-BRD1X

3.3.5 SPI host connection

Host controller SPI leader can be connected to EVSE-SIG-BRD1X SPI follower using one of the following options:

- Arduino connector J36 pinout and Arduino connector J39 pinout
- EXP CN connector J44 pinout
- MFP connector J45 pinout

3.3.6 UART host connection

Host controller UART serial port can be connected to EVSE-SIG-BRD1X UART using one of the following options:

- Arduino connector J40 pinout
- EXP CN connector J44 pinout
- MFP connector J45 pinout

3.3.7 Ethernet host connection

Host controller Ethernet port can be connected to EVSE-SIG-BRD1X using one of the following options:

- RJ45 connector P4 (100BASE-TX)
- J34 connector (100BASE-T1)

3.3.8 Auxiliary UART

An auxiliary UART port can be connected to connector header J46. The host controller can communicate to another UART peripheral through this UART port when the command requests arriving from host connectors is passed through the LPC5536/LPC55S36.

4 EVSE-SIG-BRD1X EV configuration

4.1 Host controller S32G2-VNP-RDB2

Figure 14 shows connection diagram of S32G-VNP-RDB2 host board with EVSE-SIG-BRD1X.

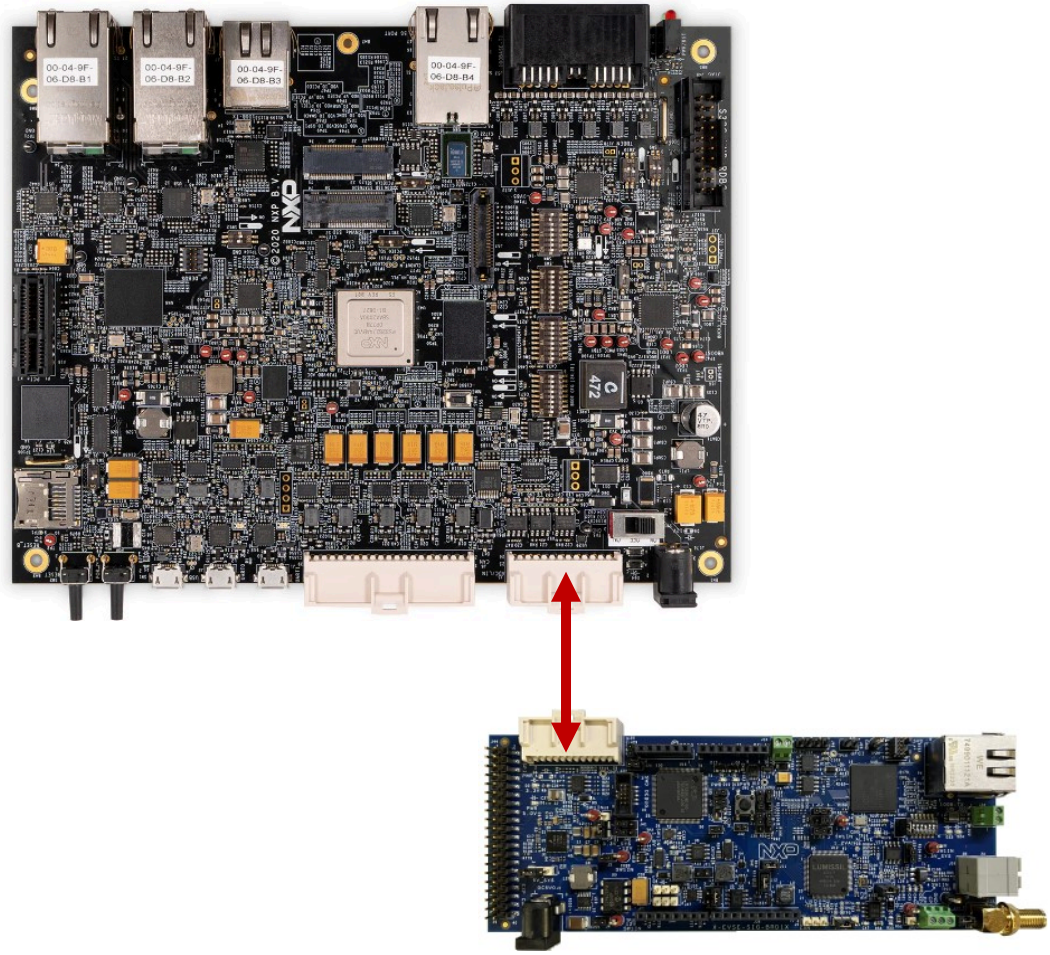


Figure 14. Connection between S32G2-VNP-RDB2 and EVSE-SIG-BRD1X

4.2 EV configuration

4.2.1 Proximity pilot sense

To use the proximity pilot hardware of the board for the EV simulation case, some modifications are required as follows:

1. Populate a 0 Ω resistor at R55.
2. Move J10 from default position 1 and 2 to position 2 and 3.

Note: *This configuration can be the same as for the EVSE simulation, see* *Section 3.3.1 "Proximity pilot sense"*.

4.2.2 Control pilot

In the EV simulation, the jumper J8 pins 3 and 4 must be short to route the PWM signal to the terminal connector J9 pin 2.

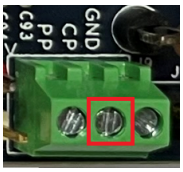
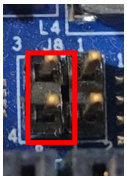
EV states can be changed to state C or D, as per J1772, either manually or via MCU GPIO control as follows:

- For MCU-controlled state switching, use header J48 pins 1 and 2. These pins must be short for state C indication, that is, charging without ventilation. Pins 2 and 3 of J48 must be short together for state D indication, that is, charging with ventilation.

- For manually controlled state switching, use header J47 pins 1 and 2. These pins must be short for state C indication, that is, charging without ventilation. Pins 2 and 3 of J47 must be short together for state D indication, that is, charging with ventilation.

The HPGP CG5317 on board does the ISO 15118 signal and data communication. To combine the J1772 PWM to the ISO 15118 signal, the jumper J14 pins 1 and 2 must be short. It helps route the ISO 15118 signal input from the terminal connector J9 pin 2 to the HPGP circuit of the board.

Table 25. Connectors/jumpers for control pilot in EV setup

Terminal connector/jumper setup	Setting	Description
J9	Pin 2	 J9 pin 2 is the control pilot signal in the terminal
J8	Pins 3 and 4 short	 Short pins 3 and 4 for EVSE mode
J14	Pins 1 and 2 short	 Short pins 1 and 2 to route the ISO 15118 signal to the HPGP circuit.

4.2.3 SPI host connection

For details, refer Section 3.3.5 "SPI host connection".

4.2.4 UART host connection

For details, refer Section 3.3.6 "UART host connection".

4.2.5 Ethernet host connection

For details, refer Section 3.3.7 "Ethernet host connection".

4.2.6 Auxiliary UART

For details, refer Section 3.3.8 "Auxiliary UART".

5 Software development

This section describes the software requirements to get started with software development with EVSE-SIG-BRD1X.

Table 26. Required software for EVSE-SIG-BRD1X

Required software	Description	Link/how to access	Additional information or comment
IDE	MCUXpresso IDE	MCUXpresso Integrated Development Environment (IDE)	For IDE installation details, refer Section 6 "Downloading and installing MCUXpresso IDE in Windows 10".
SDK	LPC5536 SDK	MCUXpresso SDK Builder	• Download and install the LPCXpresso55S36 SDK v 2.14.0 to compile version V1 of the EVSE-SIG-BRD1X projects. • SDK installation; for more details, refer Section 7 "Downloading and installing LPC5536/LPC55S36 SDK".
Base software	EVSE-SIG-BRD1X EVSE simulation software	http://www.nxp.com	Contact the NXP support team or local field application engineer (FAE)
Base software	EVSE-SIG-BRD1X EV simulation software	http://www.nxp.com	Contact the NXP support team or local field application engineer (FAE)

5.1 MCUXpresso IDE

The MCUXpresso IDE is used to edit, build, and program the provided EVSE-SIG-BRD1X MCUXpresso sample projects.

It is available for download at MCUXpresso-IDE.

For details on how to download and install the MCUXpresso IDE, using Windows OS, refer Section 6 "Downloading and installing MCUXpresso IDE in Windows 10".

5.2 EVSE-SIG-BRD1X MCUXpresso sample project

The two sample projects mentioned in this document are as follows:

- EVSE-SIG-BRD1X EVSE simulation project (LPC5536)
- EVSE-SIG-BRD1X EV simulation project (LPC5536)

Ensure to import and program the corresponding project variant for the specific EVSE-SIG-BRD1Xs for EVSE and EV simulations. For details on how to import the sample projects, refer Section 6 "Downloading and installing MCUXpresso IDE in Windows 10" and Section 7 "Downloading and installing LPC5536/LPC55S36 SDK".

5.3 MCUXpresso SDK

To build the EVSE-SIG-BRD1X EVSE/EV sample projects, the following SDK is required:

- LPC5536 EVK MCUXpresso SDK

The SDK is available for download at MCUXpresso SDK Builder.

For details on download and installation, refer Section 7 "Downloading and installing LPC5536/LPC55S36 SDK".

5.4 Programming EVSE-SIG-BRD1X software

5.4.1 Build EVSE simulation software

To build EVSE simulation software, perform the following steps:

1. Click **Import project(s) from the file system...** from the **Quickstart Panel** of MCUXpresso IDE.
2. Select the *.zip archive file and import the EVSE simulation project `evsesigbrd_sw.zip`.
3. Click the **Finish** button.

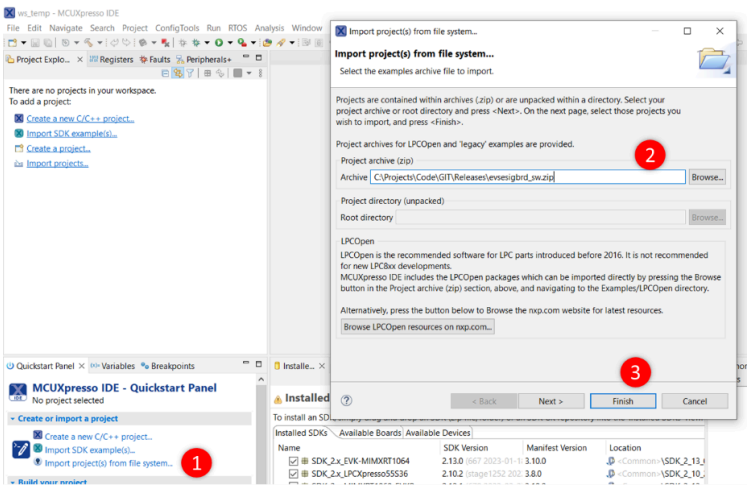


Figure 15. Import project

4. Click the  button on the top-left side of the IDE and start building the project. The build is done without errors.

5. Click the  button on the top-left side of the IDE and start programming the board with the project binary. Once the programming is completed, it breaks at a breakpoint at the `main()` function of the code.

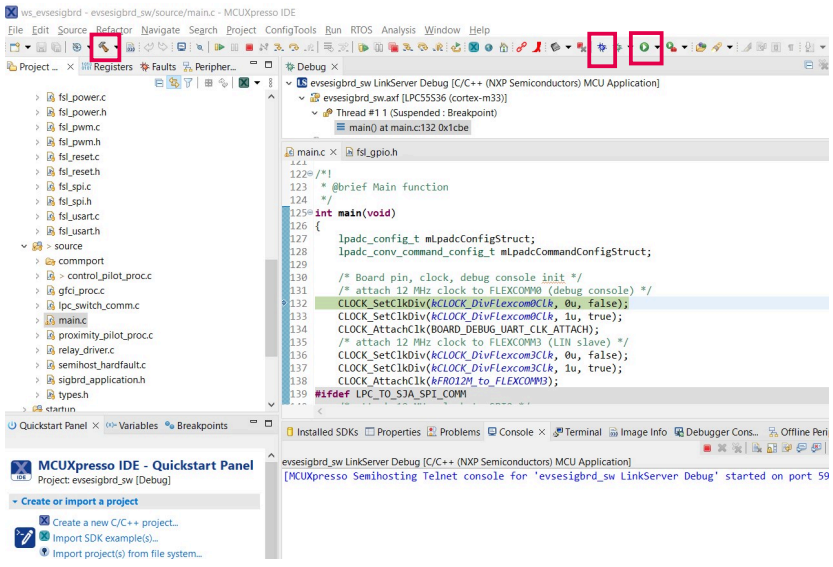


Figure 16. Project explorer

6. To resume the operation, click the  button.

5.4.2 Build EV simulation software

To build EV simulation software, perform the following steps:

1. Click **Import project(s) from the file system...** from the **Quickstart Panel** of MCUXpresso IDE.
2. Select the *.zip archive file and import the EV simulation project `evsesigbrd_sw.zip`.
3. Click the **Finish** button.

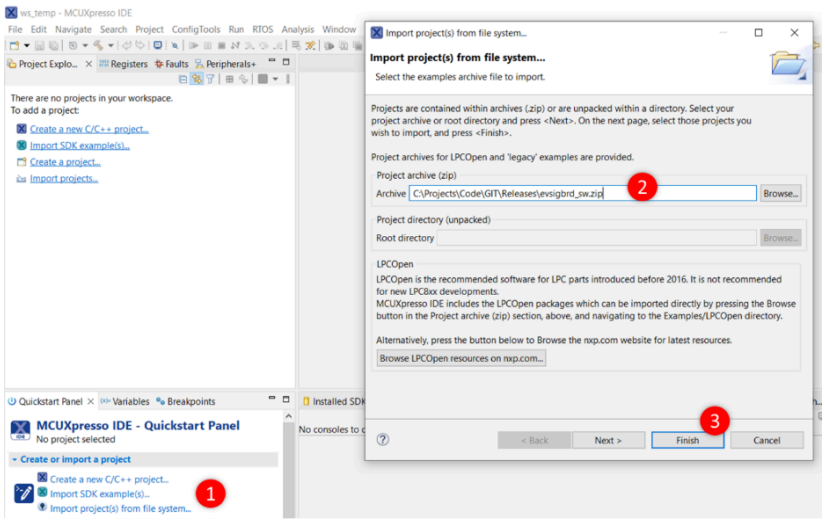


Figure 17. Import project

4. Click the  button on the top-left side of the IDE and start building the project. The build is done without errors.

5. Click the  button on the top-left side of the IDE and start programming the board with the project binary. Once the programming is completed, it breaks at a breakpoint at the `main()` function of the code.

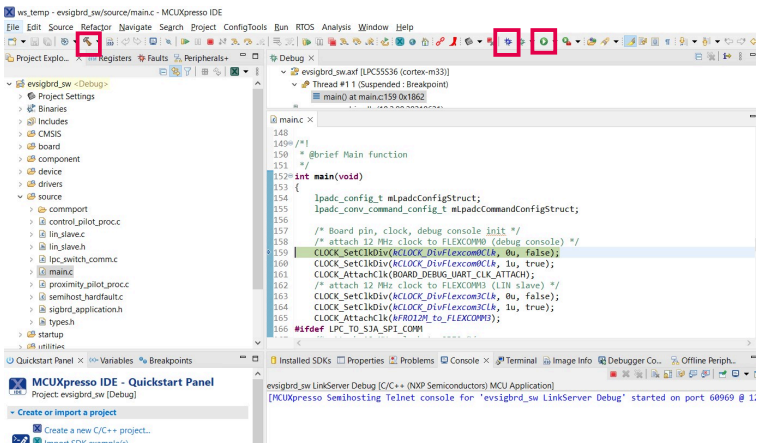


Figure 18. Project explorer

6. To resume the operation, click the  button.

6 Downloading and installing MCUXpresso IDE in Windows 10

MCUXpresso IDE is a free-of-charge, code-size-unlimited, and easy-to-use IDE for Kinetis and LPC MCUs and i.MX RT crossover processors.

To install MCUXpresso IDE, perform the following steps:

1. Go to the MCUXpresso-IDE and click the **Download** button.

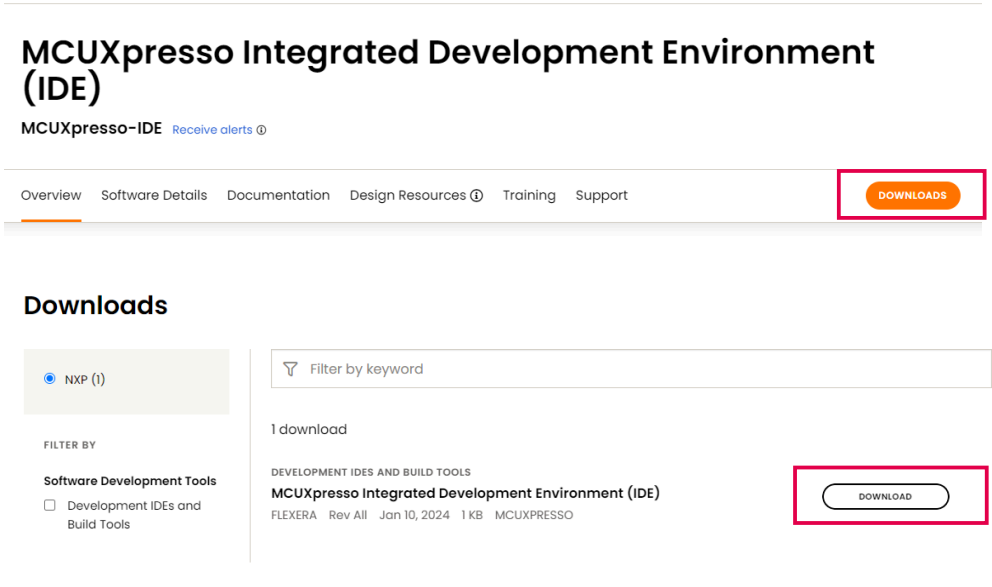


Figure 19. Download MCUXpresso IDE

2. Sign in to your account at the NXP website. If you do not have an account, click **CREATE AN ACCOUNT**.

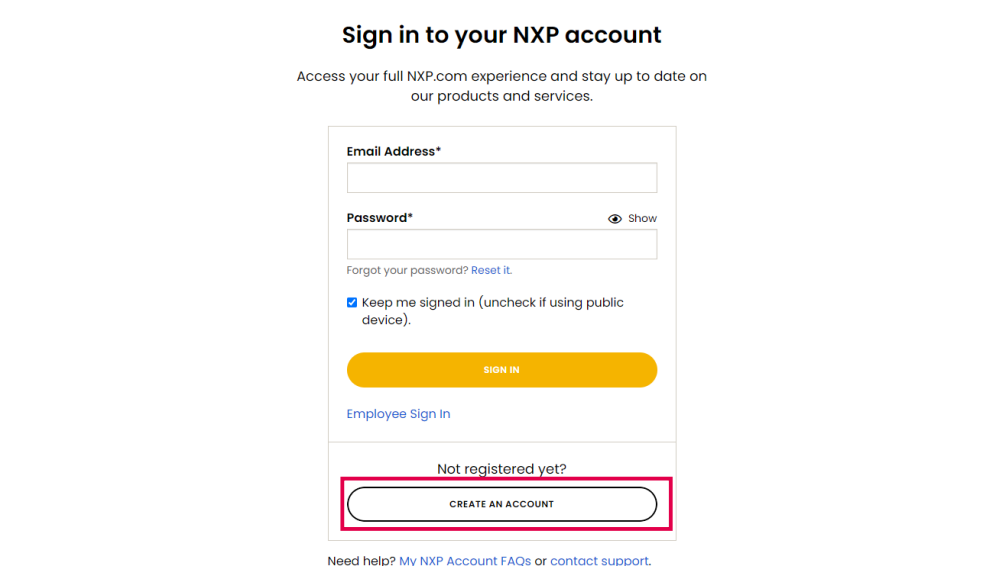


Figure 20. Creating an account

3. If you are an existing user, click **Employee Sign In** and enter your email address or NXP ID, and password.

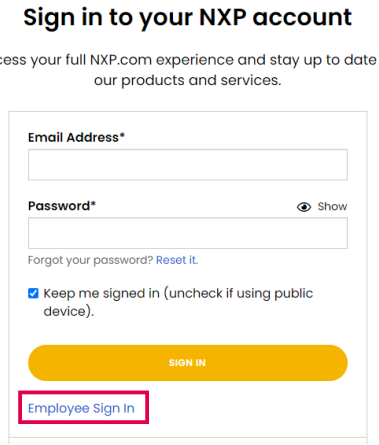


Figure 21. Signing in

4. Click MCUXpresso IDE.

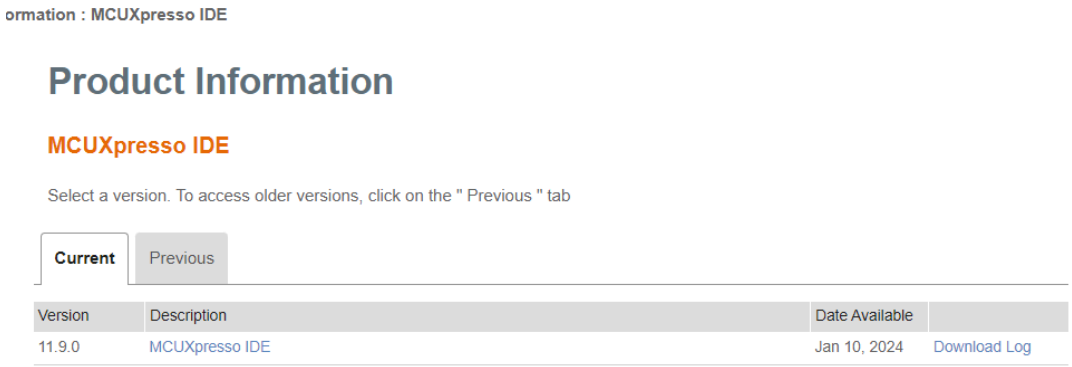


Figure 22. MCUXpresso IDE

Note: *Ensure to download the latest MCUXpresso version available.*

5. Accept the software terms and conditions.

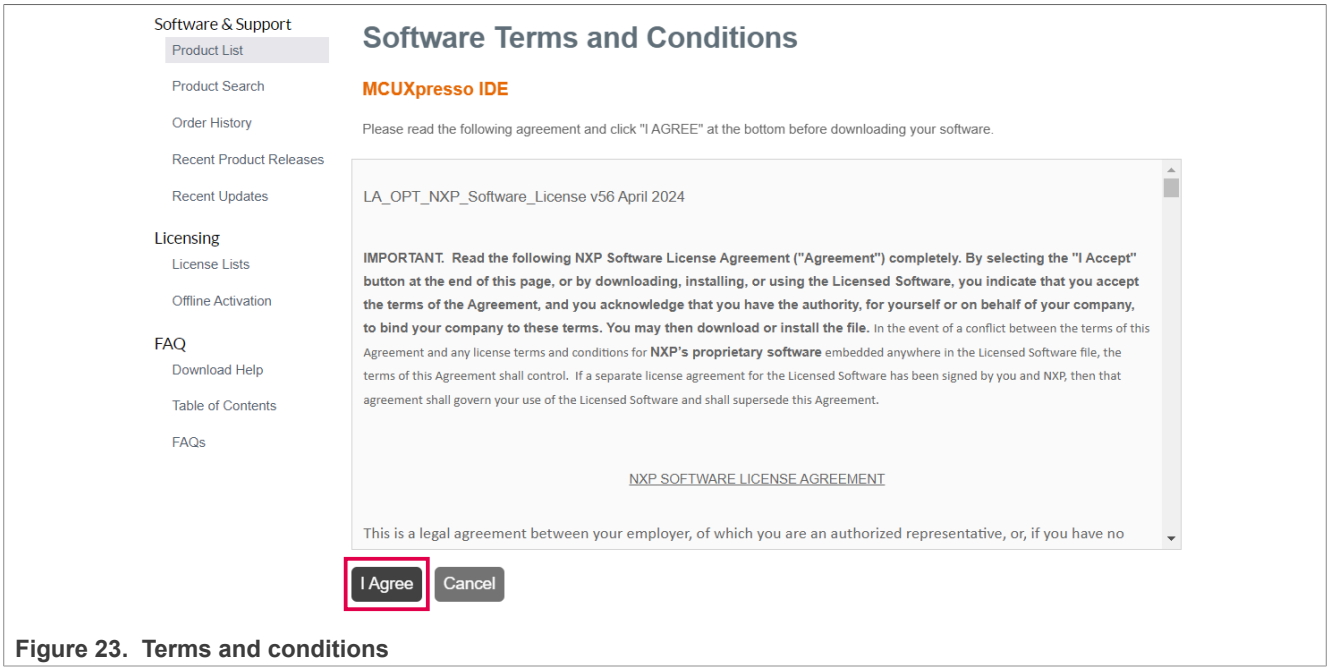


Figure 23. Terms and conditions

6. Select the MCUXpresso product version.
7. To start the download, click the corresponding **File Name**.

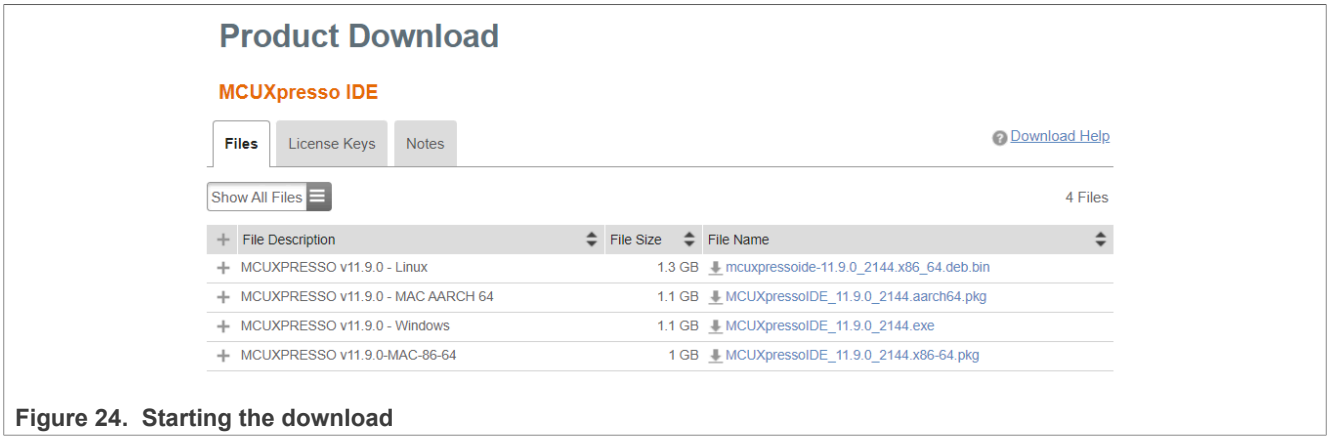


Figure 24. Starting the download

8. Double-click the installer file and follow the setup wizard until the MCUXpresso IDE installation is completed. Allow the installation of the additional drivers required by the MCUXpresso IDE during the installation process.

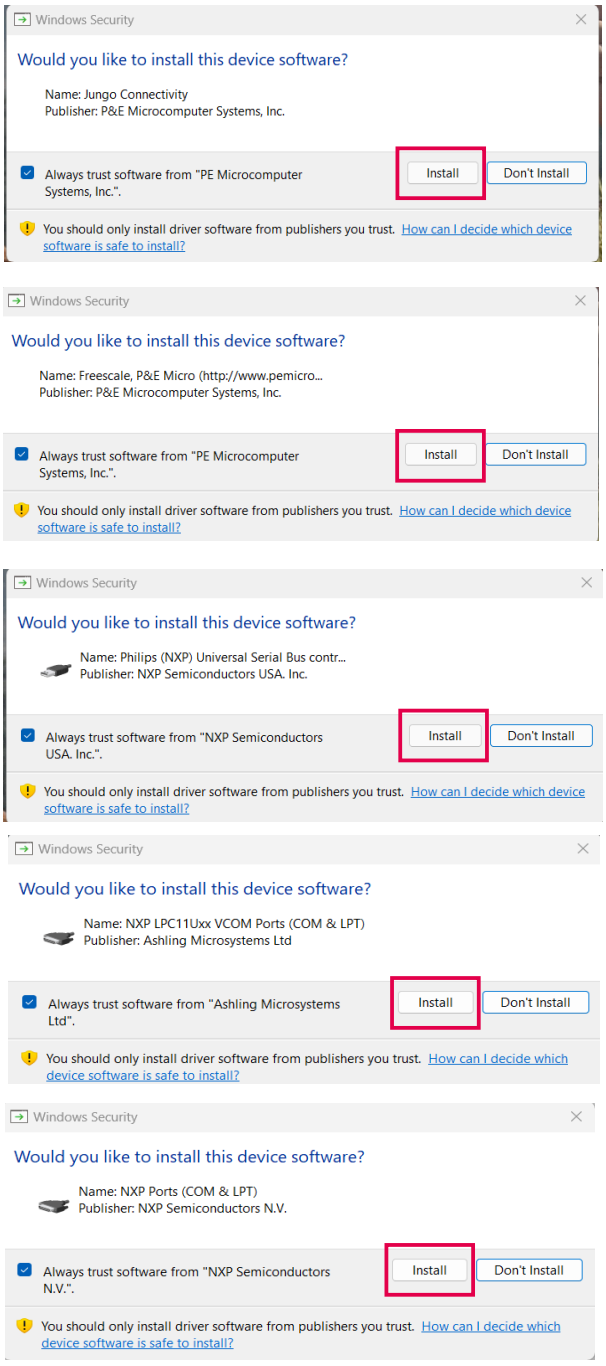


Figure 25. Device Drivers for MCUXpresso IDE

7 Downloading and installing LPC5536/LPC55S36 SDK

The following steps and illustrations show the setup processes for LPC5536-EVK, which can be used to as the SDK for LPC5536/LPC55S36 based EVSE-SIG-BRD1X.

1. Install and import the LPCXpresso55S36 SDK as follows:
   a. Browse to MCUXpresso SDK Builder and click **Select Development Board**.

b. Sign in with your NXP account. If you do not have one yet, click Register Now, enter your credentials, and click **Sign-in**.

Figure 26. MCUXpressoSDK builder

2. Enter the name of the LPCXpresso55S36 board under **Search for Hardware**.
3. Select the required board from the drop-down list and select the recommended SDK release version.
4. Click **Build MCUXpresso SDK**.

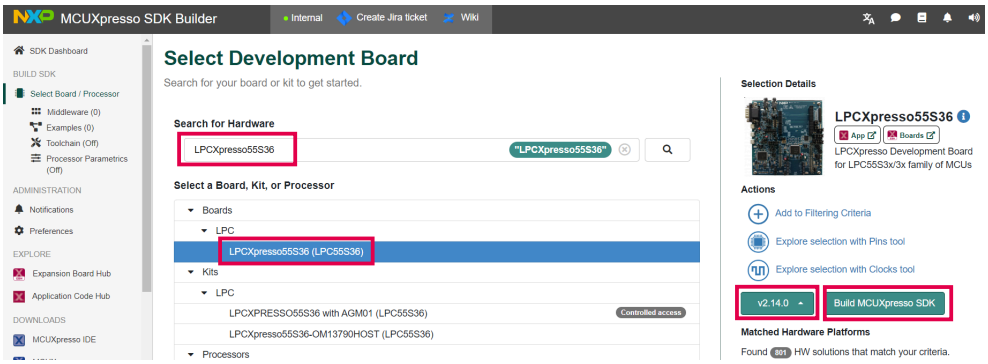


Figure 27. Select development board

Note: *Get the recommended SDK release version from section A.3 Recommended SDK release version.*

5. When building the SDK, specify the **Host OS,** and specify "MCUXpresso IDE" as the **Toolchain**. For simplicity reasons, select all the available middleware and click **Download SDK**.

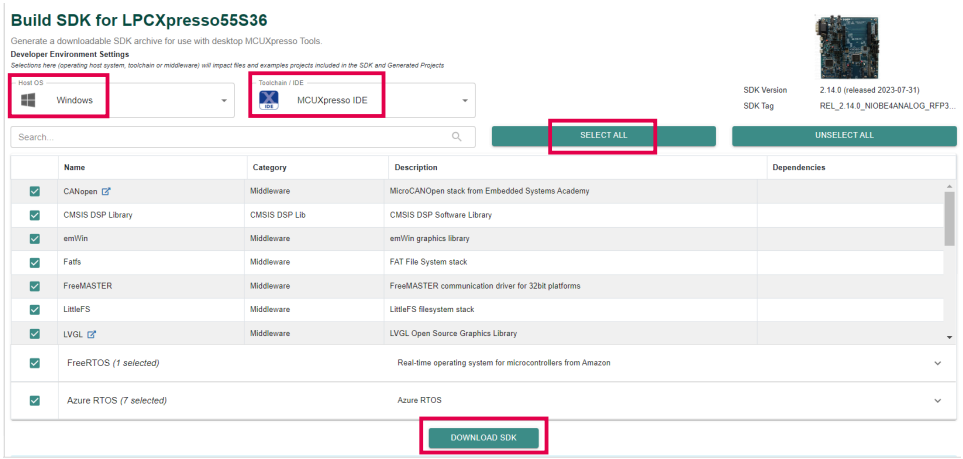


Figure 28. Build SDK for LPCXpresso55S36

6. When the build completes, download the SDK archive (9) and agree to the software terms and conditions.

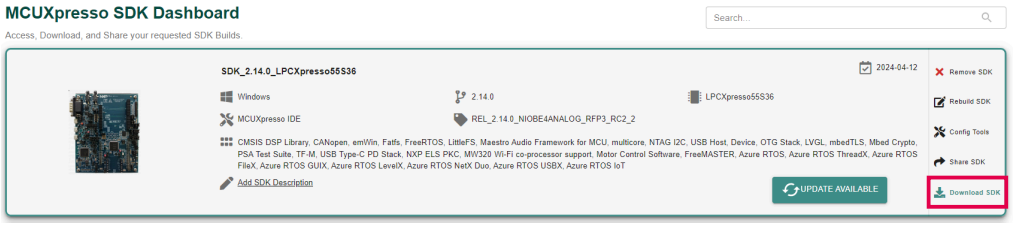


Figure 29. LPCXpresso55S36 SDK Dashboard

7. Open the MCUXpresso IDE in your desired workspace.
8. Drag and drop the SDK into the **Installed SDKs** window of the IDE.

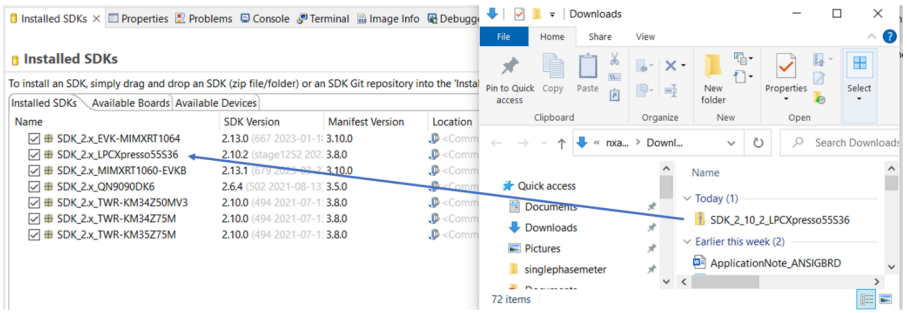


Figure 30. Install an SDK in MCUXpresso

8 Related resources

Table 27 lists some additional resources that can be required while working on the EVSE-SIG-BRD1X.

Table 27. Related resources

Resource	Link/how to obtain
EVSE-SIG-BRD1X User Manual (UM12013)	Contact an NXP field applications engineer (FAE) or sales representative
SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler (J1772_201710)	https://www.sae.org/standards/content/j1772_201710/
Lumissil website (connectivity)	https://www.lumissil.com/products/wired-communication

9 Acronyms

Table 28 lists the acronyms used in this document.

Table 28. Acronyms

Term	Description
BSD	Berkeley software distribution
CAN	Controller area network
CP	Control pilot
EV	Electric vehicle
EVSE	Electric vehicle supply equipment
GFCI	Ground-fault circuit interrupter
HPGP	HomePlug Green PHY

Table 28. Acronyms...*continued*

Term	Description
LIN	Local interconnect network
MCU	Microcontroller unit
MFP	Multifunction port
MPU	Microprocessor unit
PHY	Physical layer
PP	Proximity pilot
PWM	Pulse width modulation
SWD	Single wire debug
UART	Universal asynchronous receive transmit

10 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

11 Revision history

Table 29 summarizes the revisions done to this document.

Revision history

Revision number	Release date	Description
UG10109 v.1.0	18 June 2024	Initial public release

Legal information

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.

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.

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

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 https://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.

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.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

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.

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.

Contents