UM3269

User manual

Getting started with the AEK-POW-SPSB081 automotive power management IC evaluation board with LIN and CAN-FD

Introduction

Embedded application processors and components in complex systems require supply with multiple power rails and multiple power domains. Power management integrated circuits (PMICs) integrate multiple voltage regulators and control circuits into a single chip, thus representing an excellent option to implement complete and integrated power supply solutions. Designing with PMICs simplifies power management in complex embedded systems, optimizes power consumption and extends battery life. In addition, they reduce circuit board space and component count.

PMICs are an excellent fit for embedded processing in a wide range of applications such as body control modules, seat control modules, sunroof modules, tailgate modules, door modules, light control modules, gear shifters, steering columns, fuel pumps, electric vehicles, and autonomous vehicles.

The AEK-POW-SPSB081 is a power management IC evaluation board with enhanced power management functionalities, such as various standby modes to minimize power consumption and, featuring programmable local and remote wake-up capability.

The board is based on the SPSB081 power management system IC, which embeds one low-drop voltage regulator (V1) to supply the system microcontroller and another voltage regulator (V2) to supply external peripheral loads such as sensors.

V1 has a fixed rail of 5 V and features overvoltage detection and protection solutions, while V2 works in two different ways: as a voltage tracker of V1, or as an independent voltage regulator programmable via SPI with 5 V or 3.3 V.

Four additional high-side drivers allow supplying and driving LEDs and sensors. These high-side drivers are driven via SPI and can be configured in four different modes: OFF, ON, TIMED (with programmable time), and PWM (configurable via device registers).

Moreover, you can set the high-side driver outputs to be driven by the DIR pin. This functionality allows the user to generate custom PWM signals for the high-side outputs.

All outputs are short-circuit protected and able to detect open-load.

The communication protocol used to configure SPSB081 registers is the SPI, implemented with four wires (MISO, MOSI, CSN, and CLK).

The AEK-POW-SPSB081 also exploits embedded LIN and CAN transceivers, allowing the board to bridge between the microcontroller and the CAN and LIN communication lines.

An external microcontroller (for example, AEK-MCU-C1MLIT1) has to refresh periodically a watchdog TRIG bit in the SPSB081 register via SPI to maintain the device in active mode. In case of watchdog failure, the device enters the V1_standby (for energy saving). To wake it up, send a pulse to the WU1_IN pin or just press the S1 button. By placing a jumper on JP1, the device enters the debug mode where the watchdog is inactive.

Three demos are available in the AutoDevKit ecosystem, each of them based on a different SPC58 microcontroller evaluation board connected with an AEK-POW-SPSB081.

The demos show how to use the outputs, configuring them in four different modes: OUT 1 for ON, OUT 2 for TIMED, OUT 3 for PWM and OUT 4 for DIR.

V2 is configured as a linear regulator and changes settings alternatively every 2.5 s between 3.3 V and 5 V.

CAN connector and CAN_rx/CAN_tx pins are connected to the microcontroller board.

The CAN test signals transmitted from the microcontroller every five seconds can be effectively decoded through CAN_H/CAN_L pins.

Warning: *The AEK-POW-SPSB081 evaluation board has not to be used in a vehicle as it is designed for R&D laboratory use only.*

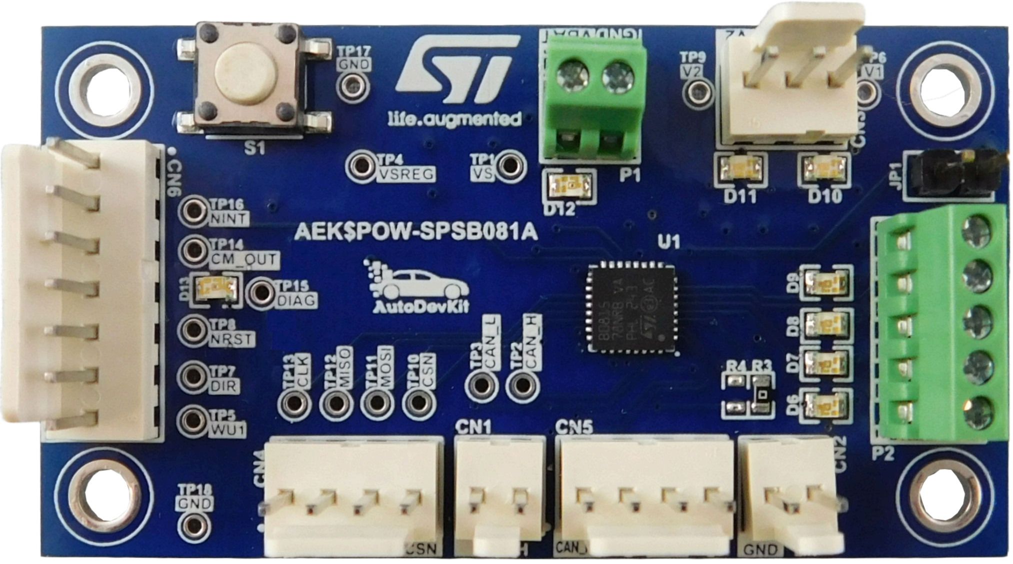


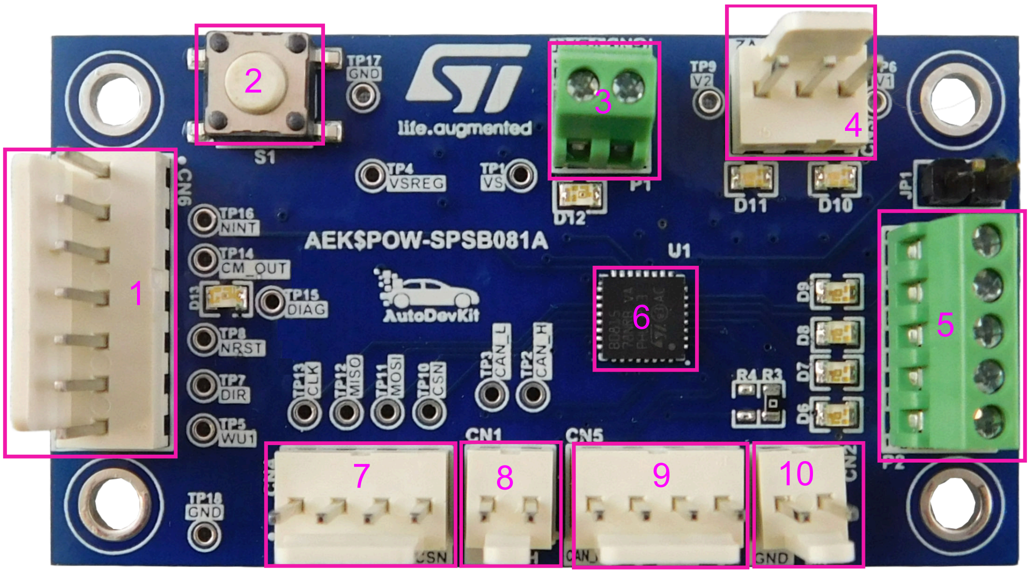
Figure 1. AEK-POW-SPSB081 evaluation board

1 Hardware overview

1.1 Board main components

1. The CN6 connector hosts the following pins (from top to bottom): GND, NINT, CM_OUT, DIAG, NRST, DIR, W1
2. Wake-up button
3. P1 connector (GND and VBAT)
4. CN3 connector (V2, GND, and V1)
5. P2 connector hosts (from top to bottom): GND and four outputs (OUT1, OUT2, OUT3, and OUT4)
6. SPSB081 automotive-grade PMIC
7. SPI connector
8. CAN-FD connector
9. CN5 connector with two CAN Tx/Rx pins (on the left side) and two LIN Tx/Rx pins (on the right side)
10. LIN connector

Figure 2. AEK-POW-SPSB081 board components



Note: In the CN4 connector for SPI, the second and third pin labels have been inverted. The correct label for the second pin is MISO, while the correct label for the third pin is MOSI.

1.1.1 SPSB081

The SPSB081 is a power management system IC providing electronic control modules with enhanced power management functionality, including various standby modes to minimize the power consumption with programmable local and remote wake-up capability, as well as LIN and CAN FD physical communication layers. The device has one low-drop voltage regulator to supply the system microcontroller and one voltage tracker to supply external peripheral loads such as sensors. V1 is available with a fixed rail (5 V or 3.3 V) and V1 overvoltage detection and protection solution, while V2 is a tracker voltage regulator of V1, programmable by SPI with 5 V or 3.3 V. Moreover, the device features four high-side drivers to supply LEDs and sensors. All outputs are short-circuit (SC) protected and implement open-load diagnosis. The STMicroelectronics standard SPI interface allows control and diagnosis of the device and enables generic software development.

1.1.2 SPSB081 block diagram

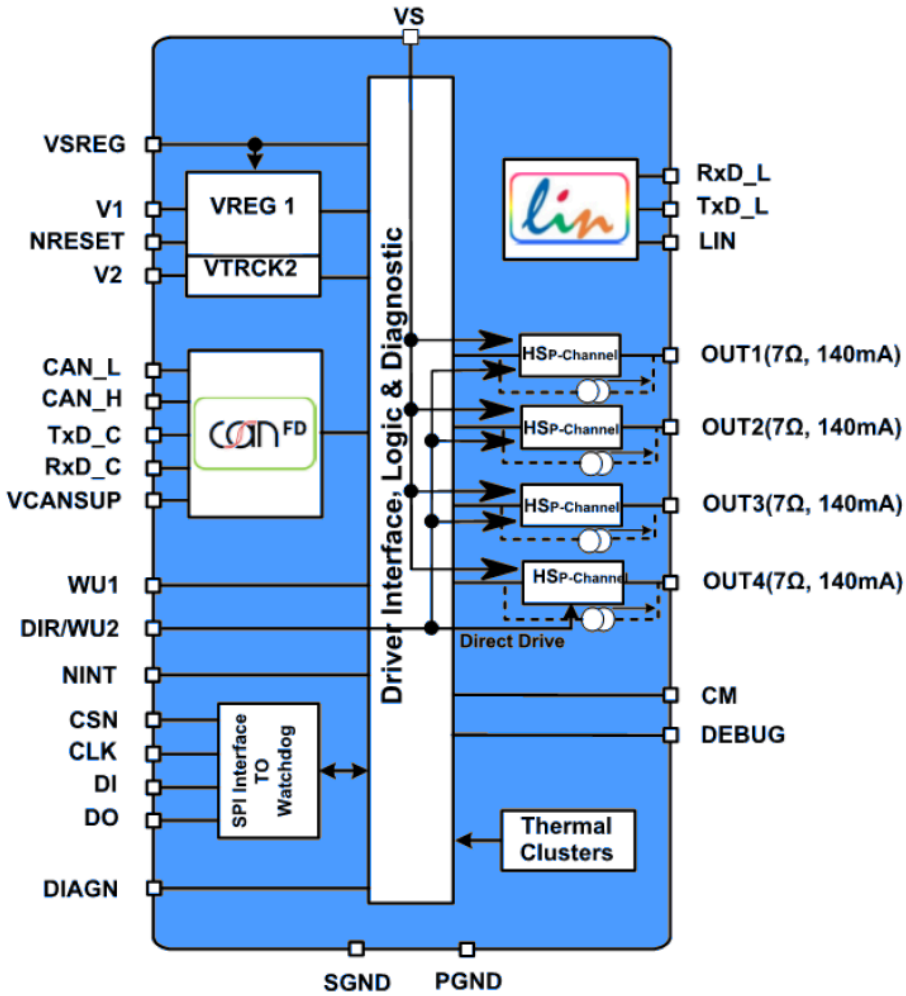


Figure 3. SPSB081 block diagram

1.1.3 Pin description and mapping on board connectors

Table 1. Board pin functions

SPBS0815	Board	Description	I/O
VBAT	P1	Board power supply (12V).	I
WU1	CN6	Wake-up inputs 1: input pin for static or cyclic monitoring of external contacts.	I
DIR	CN6	DIR → direct HS drive. By enabling DIR functionality via AutoDevKit driver APIs, when the DIR pin is driven by a GPIO of an external micro, it is possible to control one or more OUTx pins.	I
V2	CN3	Voltage regulator (5V or 3.3V) or tracker (5V): supply for external loads. When high, the D11 LED diode turns blue. V2 pin is protected against reverse supply.	O
OUT4	P2	High-side-driver output to drive LEDs, sensors or to supply contacts. When high, the D6 LED diode turns green.	O
OUT3	P2	High-side-driver output to drive LEDs, sensors or to supply contacts.	O

SPBS0815	Board	Description	I/O
		When high, the D7 LED diode turns green.	
OUT2	P2	High-side-driver output to drive LEDs, sensors or to supply contacts. When high, the D8 LED diode turns green.	O
OUT1	P2	High-side-driver output to drive LEDs, sensors or to supply contacts. When high the D9 LED diode turns green.	O
CAN_TX	CN5	CAN transmits data input.	I
LIN_TX	CN5	LIN transmits data input.	I
CM_OUT	CN6	Current monitor output: depending on the selected multiplexer bits of the control register, this output gives an image of the instant current through the corresponding high-side driver with a fixed ratio (VCM=(Ioutx/990) * Rsense).	O
DIAG	CN6	Logical error indication output. Active low.	O
NINT	CN6	Indicates device errors, warning, and local/remote wake-up events.	O
CAN_RX	CN5	CAN receives data output (push-pull output stages).	O
LIN_RX	CN5	RxDL → LIN receives data output.	O
CLK	CN4	SPI: serial clock input.	I
MOSI	CN4	SPI: serial data output (push pull output stage).	O
MISO	CN4	SPI: serial data input.	I
CSN	CN4	SPI: chip select not input.	I
NRST	CN6	NReset output to microcontroller; internal pull-up of 25 kΩ typical (active low output stage).	O
V1	CN3	Voltage regulator (5 V supply). When high, the D12 LED diode turns blue.	O
CAN_L	CN1	CAN low level voltage I/O.	I/O
CAN_H	CN1	CAN high level voltage I/O.	I/O
LIN	CN2	LIN bus line.	I/O
GND	P1, P2, CN2, CN6	Ground pin.	P

1.2 Board operating range

The AEK-POW-SPSB081 has been designed and tested with VBAT = 12 V. Even if the SPSB0815 has a wider operating range, it is recommended not to exceed this threshold, to avoid LED overcharge or board overheating.

Anyway, even if the device goes beyond the operative value, it is able to recover with no damage or degradation.

2 Functional description

2.1 Voltage regulators

The SPSB081 features a fully protected low drop voltage regulator (V1) and another low dropout voltage regulator (V2), which can work as voltage tracker, or as LDO, based on its configuration.

These regulators are designed for very fast transient response and do not require electrolytic output capacitors for stability.

Table 2. Voltage regulator configuration

Configuration	P/N	V1 output voltage	V2 output voltage (linear case)	V2 output voltage (tracking case)
3	SPSB081	5.0 V	3.3 V or 5 V	V1 voltage tracker (5 V)

The voltage regulator V2 can be configured via AutoDevKit driver APIs as a linear independent LDO (V2_TKR = 0, default) or as a tracker of the V1 voltage regulator (V2_TKR = 1).

2.1.1 Voltage regulator: V1

The V1 voltage regulator provides 5 V supply voltage (configurable by OTP) and up to 250 mA continuous load current to supply the system microcontroller and the integrated CAN transceiver. The V1 regulator is embedded in the power management and fail-safe functionality of the device and operates according to the selected operating mode.

In addition, the V1 regulator supplies the device internal loads. The voltage regulator is protected against overload and overtemperature.

In case the device temperature exceeds the TSD1 threshold as defined in the device datasheet, all outputs (OUTx, V2, LIN, CAN) are deactivated except V1 regulator, which remains on. LIN and CAN transceivers are forced in "receive only" mode. Hence, the microcontroller has the possibility of interaction or error logging. If the chip temperature exceeds the TSD2 threshold (TSD2 > TSD1) according to the values defined in the datasheet, V1 is deactivated and all wake-up sources (CAN, LIN, WU1, WU2, and timer) are disabled. After tTSD, the voltage regulator automatically restarts. If the restart fails seven times in a minute, the SPSB081 is forced into VBAT_standby mode.

In case of overvoltage, an internal pulldown circuit is activated to limit the current accumulated in the external capacitor.

2.1.2 Voltage regulator: V2

The SPSB081 embeds a 5 V low dropout tracking regulator designed to provide an output voltage that closely tracks (±20 mV) the V1 reference input while delivering up to 100 mA on the V2 output pin. The V2 voltage regulator can be configured through the V2_TKR bit in CR2 as a linear independent LDO (V2_TKR = 0, default) or as a tracker of the V1 voltage regulator (V2_TKR = 1).

The V2 can be configured as a tracker of the V1 only at the same voltage output level (5 V for SPSB081). The tracking regulator is protected against overload, overtemperature, short-circuit (short to ground and battery supply voltage), and reverse biasing.

2.1.3 Voltage regulators failure

The output voltages of V1 and V2 regulators are monitored.

In case of drop below the V1, V2 fail thresholds (V1,2 < V1, 2FAIL, for t > tV1,2FAIL), the failure bits V1FAIL, V2FAIL (SR2) are latched. These failure bits can be cleared by a dedicated SPI command.

2.1.4 Short to ground detection

At turn-on of V1 and V2 regulators, a short-to-GND condition is detected by monitoring the regulator output voltage.

If V1 or V2 is below the V1FAIL, (V2FAIL) thresholds for t > tV1SHORT (t > tV2SHORT) after turn-on, the SPSB081 will identify a short-circuit condition at the related regulator output and the regulator will be switched off.

In the case of V1 short-to-GND, the device enters VBAT_standby mode automatically. BFORCED_SLEEP_TSD2/ V1SC (SR1) and V1FAIL (SR2) bits are set.

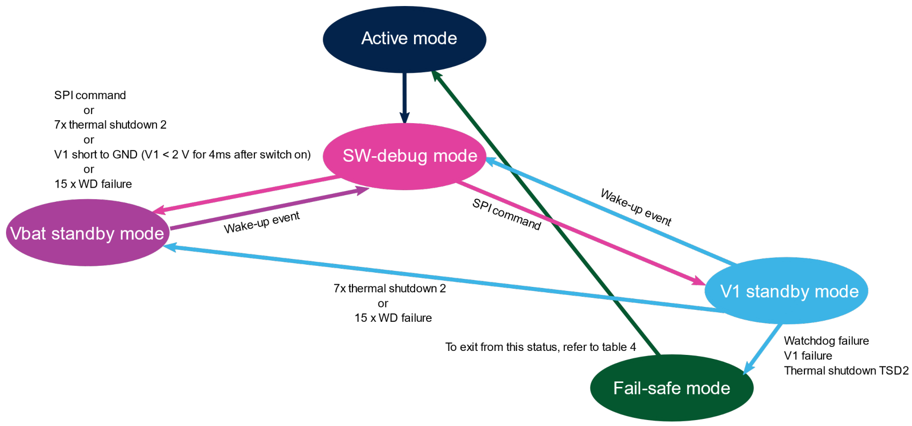
In case of a V2 short-to-GND failure, the V2SC (SR2) and V2FAIL (SR2) bits are set.

Once the output voltage of the corresponding regulator exceeds the V1FAIL (V2FAIL) threshold, the short-to-ground detection is disabled. In case of a short-to-ground condition, the regulator is switched off due to thermal shutdown.

For further details, refer to the SPSB081 datasheet available at www.st.com.

2.2 Simplified state machine

Figure 4. State machine



The AEK-POW-SPSB081 finite state machine includes the following operating modes:

- Active mode: normal operating mode
- SW-debug mode: software debugging
- V1_standby mode: low power mode
- Fail-safe-mode: safety mode triggered by failures
- VBAT_standby mode: ultralow power mode

2.2.1 Active mode

All functionalities are available. The AEK-POW-SPSB081 can be controlled via SPI.

2.2.2 SW-debug mode

To enter this mode, close the JP1 jumper. When the FSM enters this mode, the watchdog is disabled.

Note: *In this mode, you can exploit all the potentialities of the debugger application, by adding views on all the variables. It is possible to display in real-time how register values change during runtime, without compromising the operative flow of the running code due to watchdog failures.*

2.3 Fail-safe mode

2.3.1 Temporary failures

SPSB081 enters fail-safe mode in case of:

- Watchdog failure
- V1 failure (V1 < VRTX for t > tV1FS)
- Thermal shutdown TSD2

The fail-safe functionality is also available in V1_Standby mode. During V1_Standby mode the fail-safe mode is entered in the following cases:

- Watchdog failure (if watchdog still running due to IV1 > ICMP)
- V1 failure (V1 < VRTX for t > tV1FS)

- Thermal shutdown TSD2

In fail-safe mode, the device returns to a fail-safe state. The fail-safe condition is indicated to the system in the global status byte. The conditions during the fail-safe mode are:

- All outputs and V2 are turned off
- All control registers are set to default values
- Write operations to control registers are blocked until the fail-safe condition is cleared (see table 4). Only the following bits are not write-protected:
- – CR1 (0x01):
  - ◦ TRIG
  - ◦ CAN_ACT
- CR2 (0x02):
  - Timer settings (bits 8…13 and bits 16…21)
- CR5 (0x05):
  - OUT1_x (bits 0…2)
  - OUT2_x (bits 4…6)
  - OUT3_x (bits 8…10)
  - OUT4_x (bits 12…14)
- CR9 (0x09) to CR11 (0x0B)
  - PWM frequency and duty cycles
- Config register (0x3F)
  - V2_0
  - V2_1

The LIN transmitter and the CAN FD transceiver remain on. Wake-up is enabled.

If the device enters the fail-safe mode, it remains active until the fail-safe condition is removed and the fail-safe is read via SPI. The following table shows the actions to exit fail-safe mode, depending on the root cause.

Table 3. Temporary failures conditions

Failure source	Failure condition	Diagnosis	Exit from fail-safe mode
Microcontroller (oscillator)	Watchdog Early write failure or expired window	FS (global status byte) = 1 WDFAIL (SR1) = 1 WDFAIL_CNT_x (SR1) = n+1	TRIG = 1 during long open window read and clear SR1
V1	Undervoltage	FS (global status byte) = 1 V1UV (SR1) = 1(1) V1FAIL (SR2) = 1(2)	V1 > VRTX Read and clear SR1
Temperature	TJ > TSD2	FS (global status byte) = 1 TW (SR2) = 1 TSD1 (SR1) = 1 TSD2 (SR1) = 1	TJ < TSD2 Read and clear SR1

1. *V1UV bit in SR1 is set for t > tuv1 (16 µs). Fail-Safe bit GSR/FS is set only after tRD (NRESET low pulse).*
2. *If V1 < V1FAIL (for t > tV1FAIL). The fail-safe bit is in the global status register.*

2.3.2 Non recoverable failures - Forced in VBAT_standby mode

If the fail-safe condition persists and all attempts to return to normal system operation fail, the SPSB0815 is forced into VBAT_standby mode to prevent damaging the system. The VBAT_standby mode can be terminated by any wake-up source. The root cause of the VBAT_standby is indicated in the SPI status registers.

In VBAT_standby mode, all control registers are set to power on default. The VBAT_standby mode is entered in case of:

- Multiple watchdog failures: FORCED_SLEEP_WD = 1 (15x watchdog failure)
- Multiple thermal shutdowns 2: FORCED_SLEEP_TSD2/V1SC = 1 (7x TSD2)
- V1 short at turn-on (V1 < V1FAIL for t > tV1SHORT): FORCED_SLEEP_TSD2/V1SC (SR 1) = 1
- Loss of ground: SGNDLOSS(SR6) = 1

Table 4. **Non-recoverable failure conditions**

Failure source	Failure condition	Diagnosis	Exit from fail-safe mode
Microcontroller (oscillator)	15 consecutive watchdog failures	FS (global status byte) = 1 WDFAIL (SR 1) = 1 FORCED_SLEEP_WD (SR 1) = 1	Wake-up TRIG = 1 during long open window read and clear SR1
V1	Short at turn-on	FS (global status byte) = 1 V1FAIL = 1 FORCED_SLEEP_TSD2/V1SC (SR 1) = 1	Wake-up Read and clear SR1
V1	Overvoltage	FS (global status byte) = 1 FORCED_SLEEP_V1OV (SR2) = 1	Wake-up Read and clear SR1
Temperature	7 times TSD2	FS (global status byte) = 1 TW (SR 2) = 1 TSD1 (SR 1) = 1 TSD2 (SR 1) = 1 FORCED_SLEEP_TSD2/V1SC (SR 1) = 1	Wake-up Read and clear SR1
SGND	Loss of ground at SGND pin	FS (global status byte) = 1 SGNDLOSS (SR 6) = 1	Wake-up Read and clear SR1

2.3.3 V1_standby mode

The transition from active mode to V1_standby mode is controlled via SPI (STBY_SEL bit and GO_STBY bit in CR2).

To supply the microcontroller in a low-power mode, the V1 voltage regulator remains active.

After the V1_standby command (CSN low to high transition), the device enters V1_standby mode immediately and the watchdog starts a long open window (tLW). The watchdog is deactivated as soon as the V1 load current drops below the ICMP threshold (IV1 < ICMP).

The V1 load current monitoring can be deactivated by setting ICMP = 1. In this configuration, the watchdog is deactivated upon transition into V1_standby mode without monitoring the V1 load current.

LIN and CAN transmitters are switched off.

OUT1, OUT2, OUT3, OUT4, and V2 voltage regulators remain in the configuration programmed before the standby command.

Even though LIN and CAN transceivers are switched off, they can still wake up the microcontroller.

2.3.4 VBAT_standby mode

The transition from active mode to VBAT_standby mode is initiated by an SPI command (STBY_SEL bit and GO_STBY bit in CR2).

In VBAT_standby mode, V1 and V2 voltage regulators, the power outputs as well as LIN and CAN transmitters, are switched off.

A NReset pulse is generated and last asserted low upon wake-up from VBAT_standby mode and until 2 ms after LDO1 is fully operative.

2.3.5 Wake-up from standby modes

A wake-up from standby mode switches the device to active mode. This can be initiated by one or more of the events listed in the table below.

Table 5. **Wake-up source**

Wake-up source	Description
LIN bus activity	It can be disabled by SPI
CAN bus activity	It can be disabled by SPI
Level change of WU	It can be configured or disabled by SPI

Wake-up source	Description
$I_{V1} > I_{CMP}$	The device remains in V1_standby mode but the watchdog is enabled (if ICMP = 0). No interrupt is generated
Timer interrupt/ wake-up of µC by TIMER	Programmable by SPI: • V1_standby mode: The device wakes up and an interrupt signal is generated at NINT when the programmable time-out has elapsed • VBAT_standby mode: The device wakes up after the programmable timer expiration. V1 regulator is turned on and NReset signal is generated when the programmable time-out has elapsed
SPI access	Always active (except in VBAT_standby mode) Wake-up event: CSN falling edge

To prevent the system from a deadlock condition (no wake-up from standby possible), a configuration where all wake-up events are disabled, is not allowed.

Accordingly, if the following condition is set by SPI:

- CAN_WU_EN = 0
- LIN_WU_EN = 0
- WU1_EN = 0
- WU2_EN = 0

the effect is to put all these register bits in their default condition:

- CAN_WU_EN = 1
- LIN_WU_EN = 1
- WU1_EN = 1
- WU2_EN = 1

Table 6. Functional overview

Function	Comments	Operating modes		
		Active mode	V1_standby static mode (cyclic sense)	VBAT_standby static mode (cyclic sense)
Voltage regulator V1	VOUT = 5 V/3.3 V	On	On[(1)]	Off
Voltage regulator V2	VOUT = 5 V/3.3 V	On/Off[(2)]	On[(2)]/Off	On[(2)]/Off
Reset generator	-	On	On	Off
Window watchdog	V1 monitor	On	Off (On if IV1 > ICMP and ICMP = 0)	Off
Wake-up	-	Off	Active[(3)]	Active[(3)]
HS-cyclic supply	Oscillator time base	On/Off	On[(2)]/Off	On[(2)]/Off
LIN	LIN ISO 17987-4/2016	On[(2)]	Off[(4)]	Off[(4)]
CAN FD	-	On/Off[(5)]	Off[(4)]	Off[(4)]
Oscillator	-	On	On/Off[(6)]	On/Off[(6)]
VS monitor	-	On[(7)]	On/Off[(7)]	On/Off[(7)]
OUT1 (P-channel HS)	-	On/Off[(2)]	On/Off[(2)]	On/Off[(2)]
OUT2 (P-channel HS)	-	On/Off[(2)]	On/Off[(2)]	On/Off[(2)]
OUT3 (P-channel HS)	-	On/Off[(2)]	On/ Off[(2)]	On/Off[(2)]
OUT4 (P-channel HS)	-	On/Off[(2)]	On/ Off[(2)]	On/Off[(2)]
Thermal shutdown TSD2	-	On	On	Off

Function	Comments	Operating modes		
		Active mode	V1_standby static mode (cyclic sense)	VBAT_standby static mode (cyclic sense)
Thermal shutdown TSD1x (for P-channel HS)	-	On	On(8)	On/Off(2)

1. *Supply the processor in low currentmode.*
2. *According to SPI setting.*
3. *Unless disabled by SPI.*
4. *The bus state is internally stored when going to standby mode. A change of bus state will lead to a wake-up after exceeding the internal filter time (if wake-up by LIN or CAN is not disabled by SPI).*
5. *After power-on, the CAN-FD transceiver is in 'CAN Trx standby' mode. It is activated by SPI command (CAN_ACT = 1).*
6. *ON, if at least one of the following is enabled: cyclic sense, OUT1, OUT2, OUT3, OUT4, V2.*
7. *ON when at least one OUTx is enabled and OFF when all outputs are disabled.*
8. *V1, V2, OUT1, OUT2, OUT3, and OUT4 are thermal monitored.*

2.4 V1 overvoltage detection

SPSB081 protects the MCU against voltages over 5 V that could harm its functionalities. For this purpose, to sink the exceeding current accumulated in the external capacitor usually directed to the MCU, the SPSB081 uses an internal pulldown circuit, where the output current flows through the pulldown resistor to the ground. If the bit V1_OVP = 1, the overvoltage protection circuit is enabled in CR2. If V1_OVP = 0 there is no overvoltage protection anymore.

Two thresholds run the overvoltage event.

When SPSB081 detects an overvoltage event on the V1 pin:

- If $V1_{OV\_TH1} \leq V1 < V1_{OV\_TH2}$ → V1_OV status flag is raised after $t_{FILTER1}$ and at the same time NINT is pulled low for 56 µs.
- If $V1_{OV\_TH2} \leq V1 < 6.5$ V → It causes a nonrecoverable failure, and the device is forced to V_{BAT} standby mode (V1 is switched OFF, and FORCED_SLEEP_V1OV (SR2) is raised after $t_{OV1FILTER2}$). The internal pull-down transistor, which will discharge the decoupling capacitor through the $R_{DISCHARGE}$, is activated.

If the V_{BAT} stdby is caused by an output overvoltage on V1, the status flag remains set (V1_OV = 1).

2.5 Power outputs (OUT1, OUT2, OUT3, OUT4)

There are four standalone high-side outputs (OUT1, OUT2, OUT3, and OUT4) to drive LEDs, for example, or to supply contacts. These outputs can be activated in standby modes.

All high-side outputs switch off in case of:

- VS overvoltage and undervoltage
- Overcurrent (only the impacted high-side output)
- Overtemperature (TSD1)
- Fail-safe event
- Loss of ground at SGND pin

In the case of overcurrent or overtemperature (TSD1 (SR1)) condition, the drivers will switch off. The related status bit is latched and can be read and optionally cleared by SPI. The drivers remain off until the status is cleared.

In case of overvoltage and undervoltage conditions, the drivers are switched off. The relative status bit is latched and can be read and optionally cleared by SPI. If the VS_LOCK_EN bit is set to '1', the drivers remain off until the status is cleared. If the VS_LOCK_EN bit is set to '0', the drivers switch on automatically if the error condition disappears. Undervoltage and overvoltage shutdowns can be disabled by setting < VS_UV_SD_EN > respectively < VS_OV_SD_EN > to '0'. In case of open-load condition, the appropriate status register (in SR4) is set. The status can be read and optionally cleared by SPI. The high-side outputs are not switched off in case of open-load condition.

Note: The maximum voltage and current applied to the high-side outputs are specified in the 'absolute maximum ratings'. Appropriate external protection may be required in order to respect these limits under application conditions.

Each of the standalone high-side driver outputs OUT1..4 can be also driven either with an internal generated PWM signal or with an internal timer. OUT1..4 can be also directly driven with the DIR pin.

2.6 Open-load detection

The open-load detection monitors the load current in each activated output stage. If the load current is below the open- load detection threshold (IOLDx) for at least t_{FOL}, the corresponding open-load bit is set in the status register.

2.7 Overcurrent detection

In the case of an overcurrent condition, a status flag is set in the corresponding status register. If the overcurrent signal is valid for at least t_{BLK}, the overcurrent flag is set, and the corresponding driver switches off to reduce the power dissipation and to protect the integrated circuit. If the overcurrent recovery bit of the output is cleared, the microcontroller has to clear the status bits to reactivate the corresponding driver.

2.8 Current monitor

The current monitor sources a current image of the power stage output current at the current monitor pin, which has a fixed ratio of the instantaneous current of the selected high-side driver.

The reference formula is VCM=(I_OUT_x/990)*Rsense with Rsense = 27 kΩ

The signal at output CM is blanked after switching on the driver until correct settlement of the circuitry. Using the AutoDevkit driver APIs, it is necessary to select which output is multiplexed to the current monitor output CM. The current monitor output allows a more precise analysis of the actual state of the load rather than the detection of an open or overload condition. The current monitor output is enabled after the current-monitor blanking time, when the selected output is switched on. If this output is off, the current monitor output is in high-impedance mode. The current monitor can be turned on or off by selecting the corresponding setting for the CM on/off bit.

2.9 Constant current mode

For the OUT1, OUT2, OUT3, and OUT4 high-side drivers, the constant current mode (CCM) feature provides constant current to the related output. The CCM feature is configurable via SPI using AutoDevKit driver APIs; these bits can be set only if the related driver is in OFF state. When the CCM is enabled, the overcurrent and short-circuit detection of the related output is switched OFF while its open-load detection is always ON.

The CCM is automatically disabled after an expiration time tCCMTIMEOUT. The allowed sequences are:

- Set OUTx_CCM_EN bit, then turn ON the related driver by SPI (the other configurations of the OUTx_y in CR5 are ignored): driver starts in CCM for tCCMTIMEOUT, then it switches to ON mode and the OUTx_CCM_EN bit is automatically cleared;
- If OUTx_CCM_EN = 1, after driver has been started in ON, PWM, timer modes then CCM bit is ignored;
- If OUTx_CCM_EN bit is cleared by the microcontroller before timeout, then the driver is switched to ON mode.

The short-circuit and overcurrent detection are enabled in ON, PWM, timer and DIR modes, but not in constant current mode. The default value for the OUTx_CCM_EN bit is 0, that is, the CCM is disabled by default.

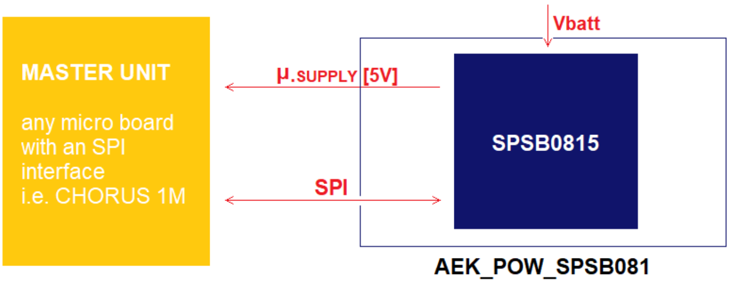
For further details on the entire section 2, refer to the SPSB081 datasheet available at www.st.com.

3 How to use the AEK-POW-SPSB081

3.1 Applications

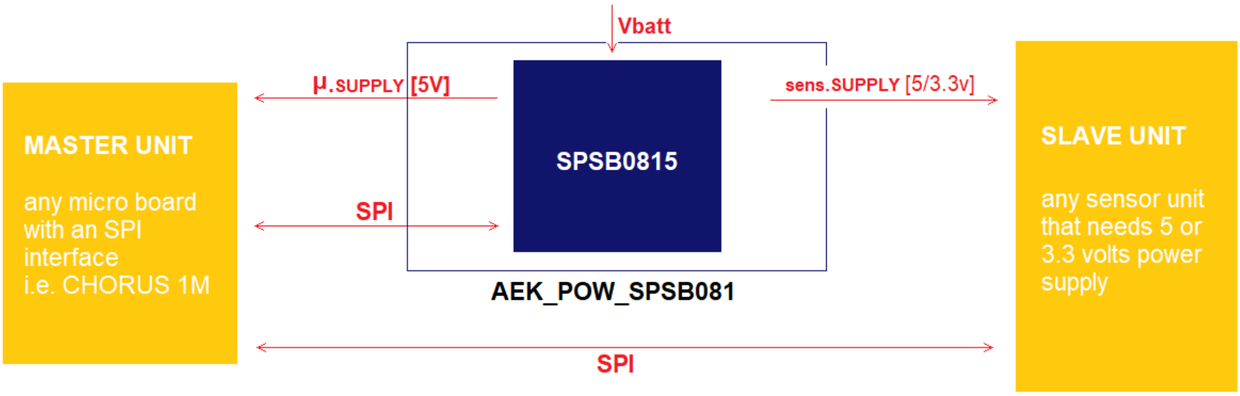
The most common application is based on the AEK-POW-SPSB081 used just to supply a microcontroller board. As soon as the AEK-POW-SPSB081 board is supplied through VBAT, V1 regulator switches to 5 V and the master unit starts exchanging SPI commands.

Figure 5. Use case 1



Another example could include a sensor board, supplied by the AEK-POW-SPSB081. When VBAT is supplied, V1 [5 V] goes high, and the master unit code starts running. In this case, the AEK-POW-SPSB081 has to be configured to switch on the V2 regulator, in order to supply the slave unit.

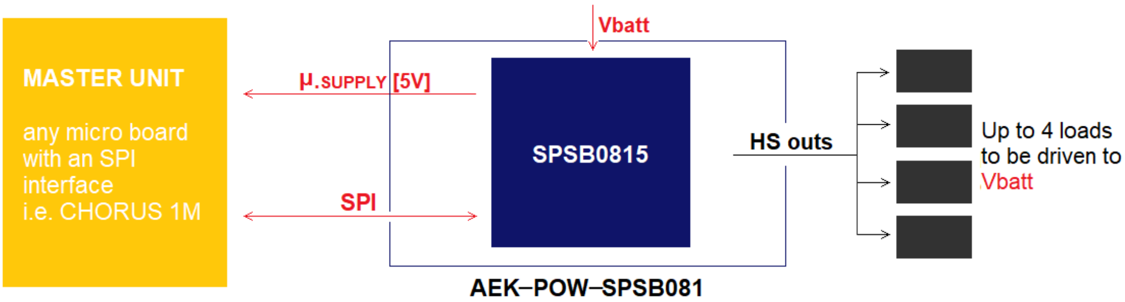
Figure 6. Use case 2



The third example exploits the capability of the board to drive external devices, which need higher currents to be driven (up to 140 mA).

Four motors can be connected to the four high-side outputs of the AEK-POW-SPSB081, and the master unit is programmed to drive them by following a predefined logic.

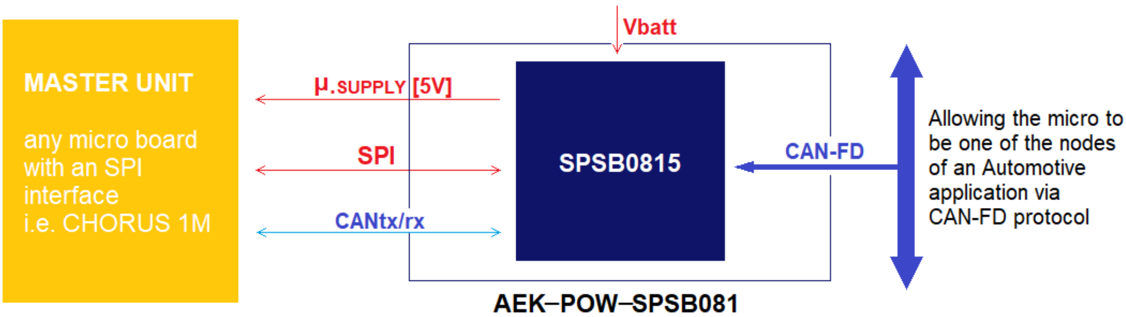
Figure 7. **Use case 3**



Another example takes advantage of the CAN transceiver embedded in the SPSB081, allowing the master to join a CAN-FD network.

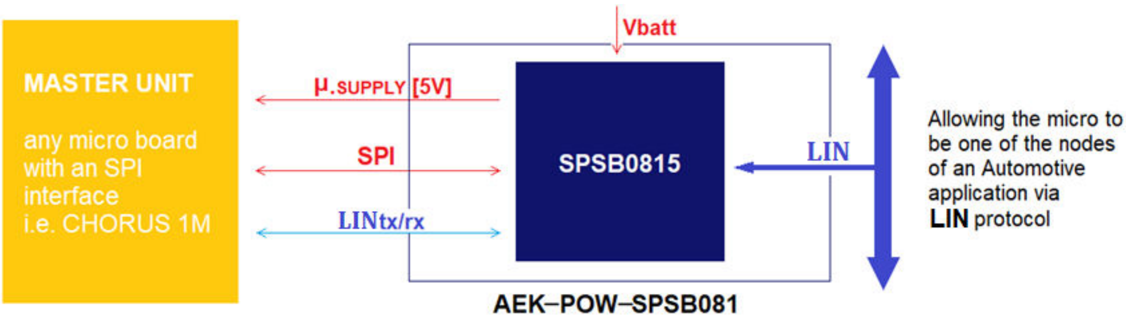
After power-on, by enabling the CAN capabilities, it is possible to exchange CAN-FD messages between the master unit and any other node of the CAN network.

Figure 8. **Use case 4**



The last example is similar to the one above but involves the LIN transceiver embedded on the SPSB081, which allows the microcontroller to communicate with a LIN network.

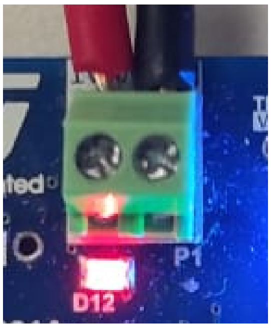
Figure 9. **Use case 5**



3.2 Using the board

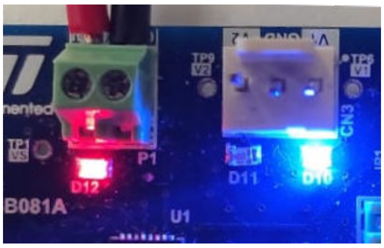
Supply the AEK-POW-SPSB081 with 12 V between VBAT/GND of P1 connector. D12 LED lights up and remains on.

Figure 10. D12 LED



At power-on, the watchdog starts a long open window. The SPSB081 enters the active mode, setting the registers at their default value. V1 regulator becomes active and D10 LED lights up in blue.

Figure 11. V1 active



Another long open window starts and the SPSB081 waits for a watchdog refresh.

Note: This function is periodically managed through a PIT (programmable timer with interrupt) in the AutoDevKit component library.

If the watchdog does not receive a refresh message of the TRIG bit (watchdog fail) 15 times in a row, the board enters the VBAT_standby mode.

Figure 12. **NRST pulses indicating watchdog failure**

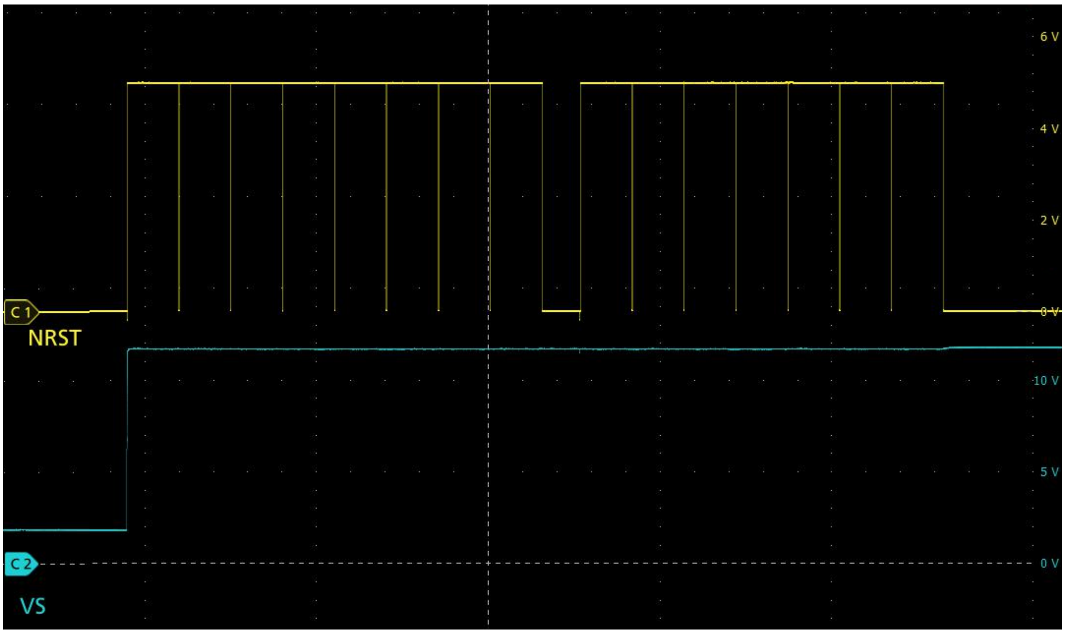
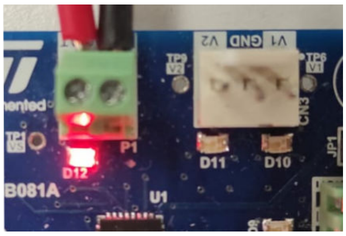


Figure 13. **D12 LED during Vbat_standby mode**



To wake up the board from this mode, press S1 button or configure a microcontroller on another board to send a high signal to WU1 pin.

Figure 14. **S1 button**

If one of the above wake-up conditions occurs, the board exits from the Vbat_standby mode to enter the active mode.

Note: If a jumper is present on JP1, the watchdog is disabled. Then, the board remains in the active mode, unless you choose an AutoDevKit library API to switch among the finite state machine modes.

4 AutoDevKit ecosystem

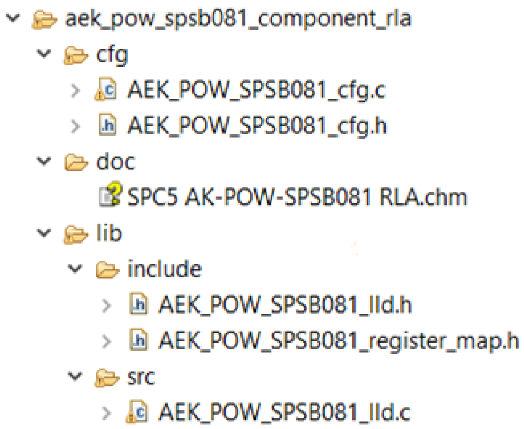
The application development employing the AEK-POW-SPSB081 takes full advantage of the AutoDevKit ecosystem, whose basic components are:

- AutoDevKit Studio IDE (STSW-AUTODEVKIT)

- PLS UDE and OpenOCD programmers and debuggers

4.1 aek_pow-spsb081_component_rla folder structure

Figure 15. **AEK-POW-SPSB081 component folder structure**



The cfg folder contains all the configuration files.

The doc folder contains the doxygen documentation.

The lib folder contains the low-level drivers of the SPSB081 and its register map.

The AEK_POW_SPSB081_lld.h contains all the APIs:

- to configure the voltage regulators (V1/V2)
- to configure the working mode of each OUT pin
- to enable/disable CAN/LIN communication
- to enable/disable wakeup sources (WU1, CAN, LIN)

4.2 Using AEK-POW-SPSB081 in AutoDevKit

In this example, we created an application for the AEK-POW-SPSB081 configured as a slave transceiver. We used the AEK-MCU-C1MLIT1 as the microcontroller board.

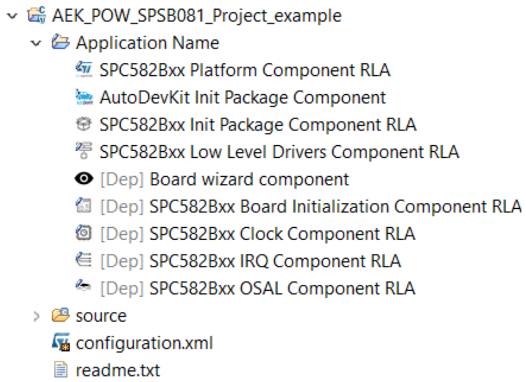
To recreate this scenario, follow the procedure below:

Step 1. Create a new SPC5-STUDIO application for the SPC582B series microcontroller and add the following components:

- SPC582Bxx Init Package Component RLA
- SPC582Bxx Low Level Drivers Component RLA
- AutoDevKit Init Package Component

These components need to be added immediately, or the other components will not be visible.

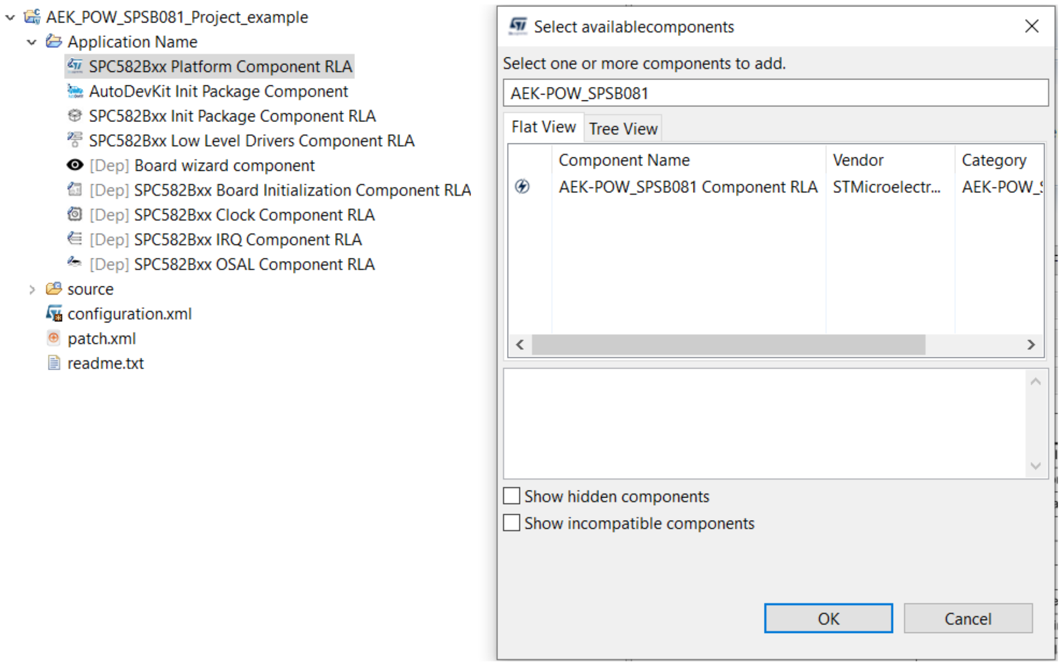
Figure 16. Adding components



Step 2. Add the following additional components:

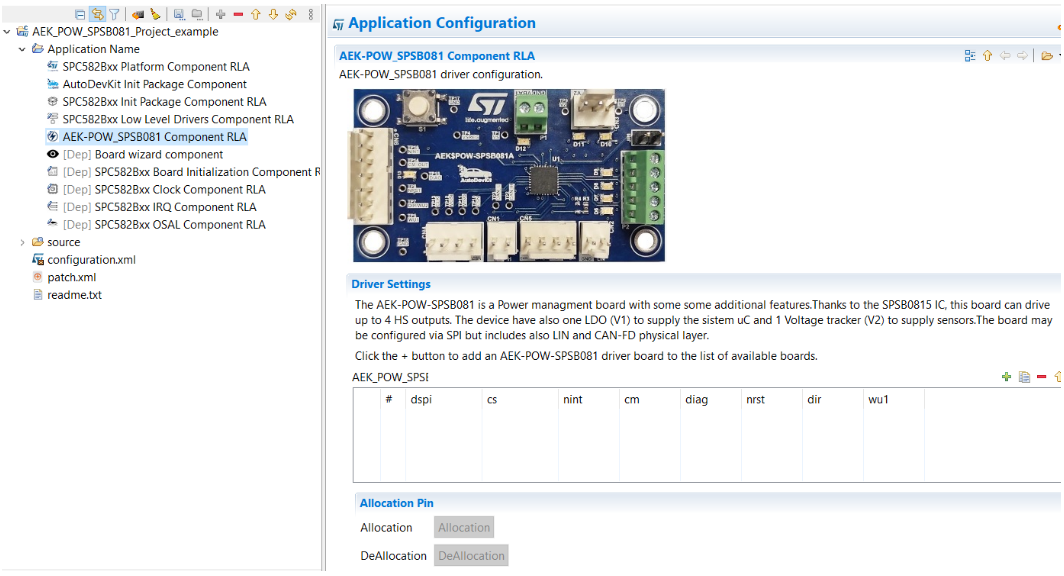
- AEK-POW-SPSB081 RLA

Figure 17. Adding AEK-POW-SPSB081 component RLA



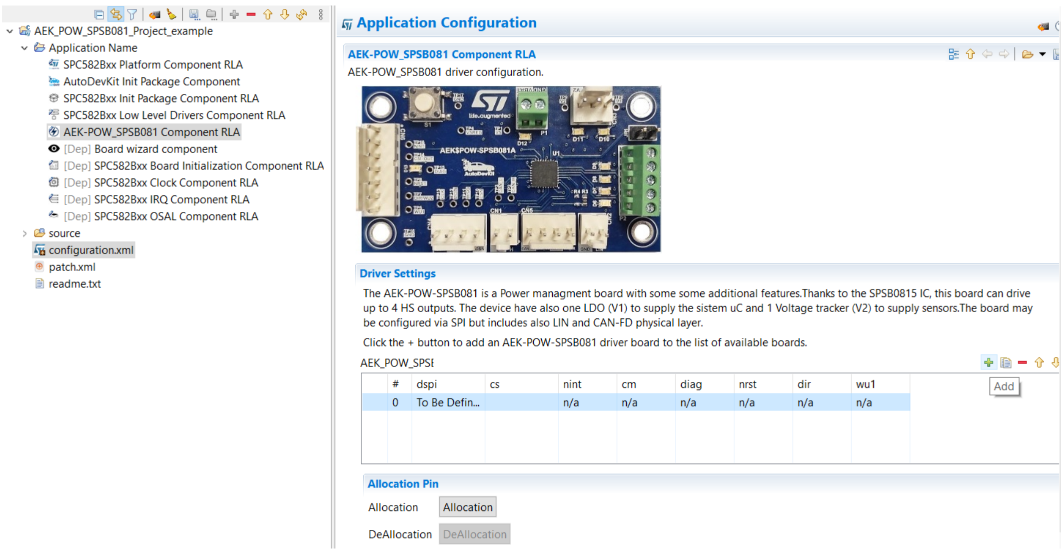
Step 3. Select [AEK-POW-SPSB081 Component RLA] to open the [Application Configuration] window.

Figure 18. Selecting AEK-POW-SPSB081 Component RLA



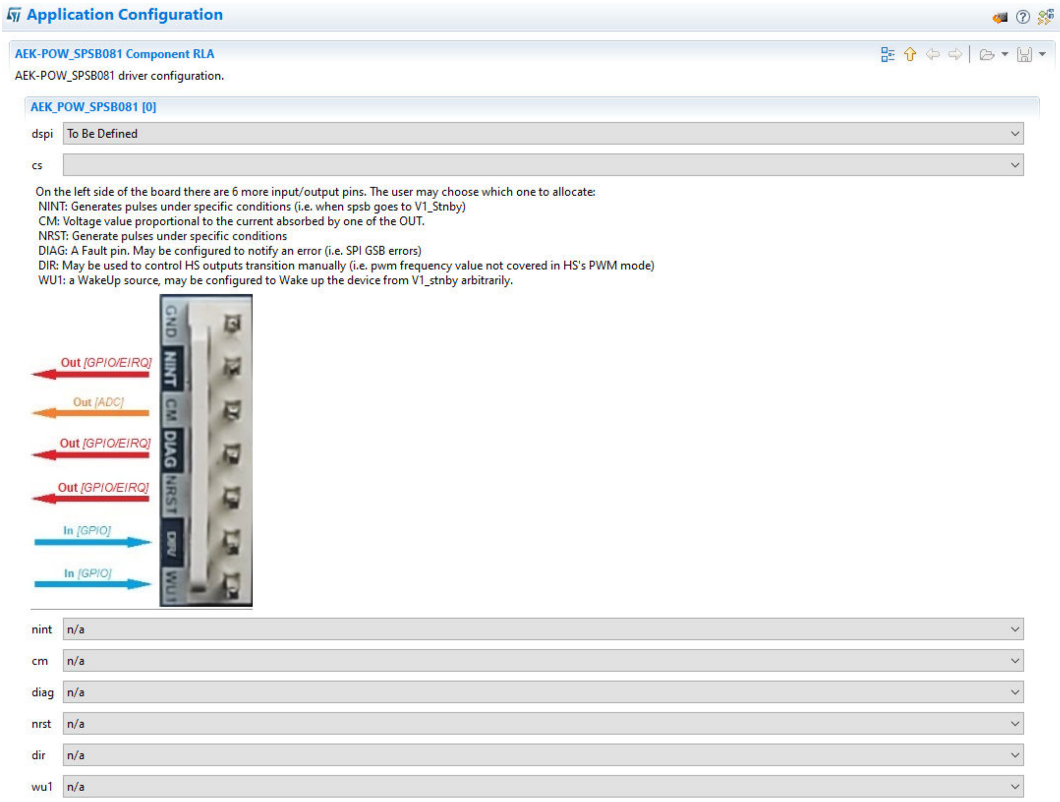
Step 4. Click on [+] to add a new element to the board list.

Figure 19. Adding a new element



Step 5. Double-click on the newly added element to configure the board.

Figure 20. AEK-POW-SPSB081 configuration



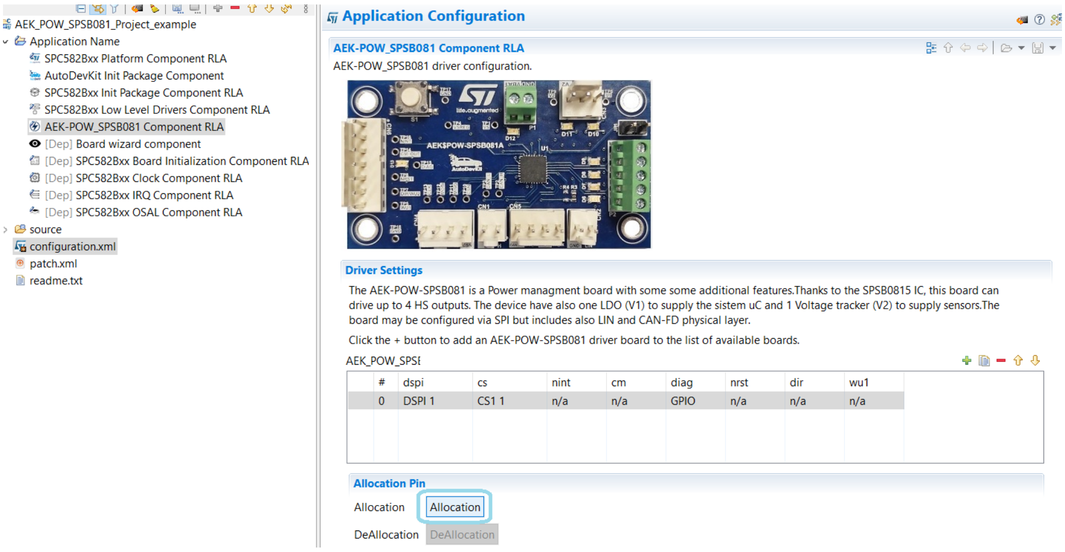
Step 6.

- Select DSPI and CS
- Select n/a, GPIO or EIRQ for NINT
- Select n/a or ADC for CM
- Select n/a, GPIO or EIRQ for DIAG
- Select n/a, GPIO or EIRQ for NRST
- Select n/a or GPIO for DIRSelect n/a or GPIO for WU1

Whenever a value different from “n/a“ is selected for the choosen pin, the allocation procedure will allocate a corresponding pin of the microcontroller board.

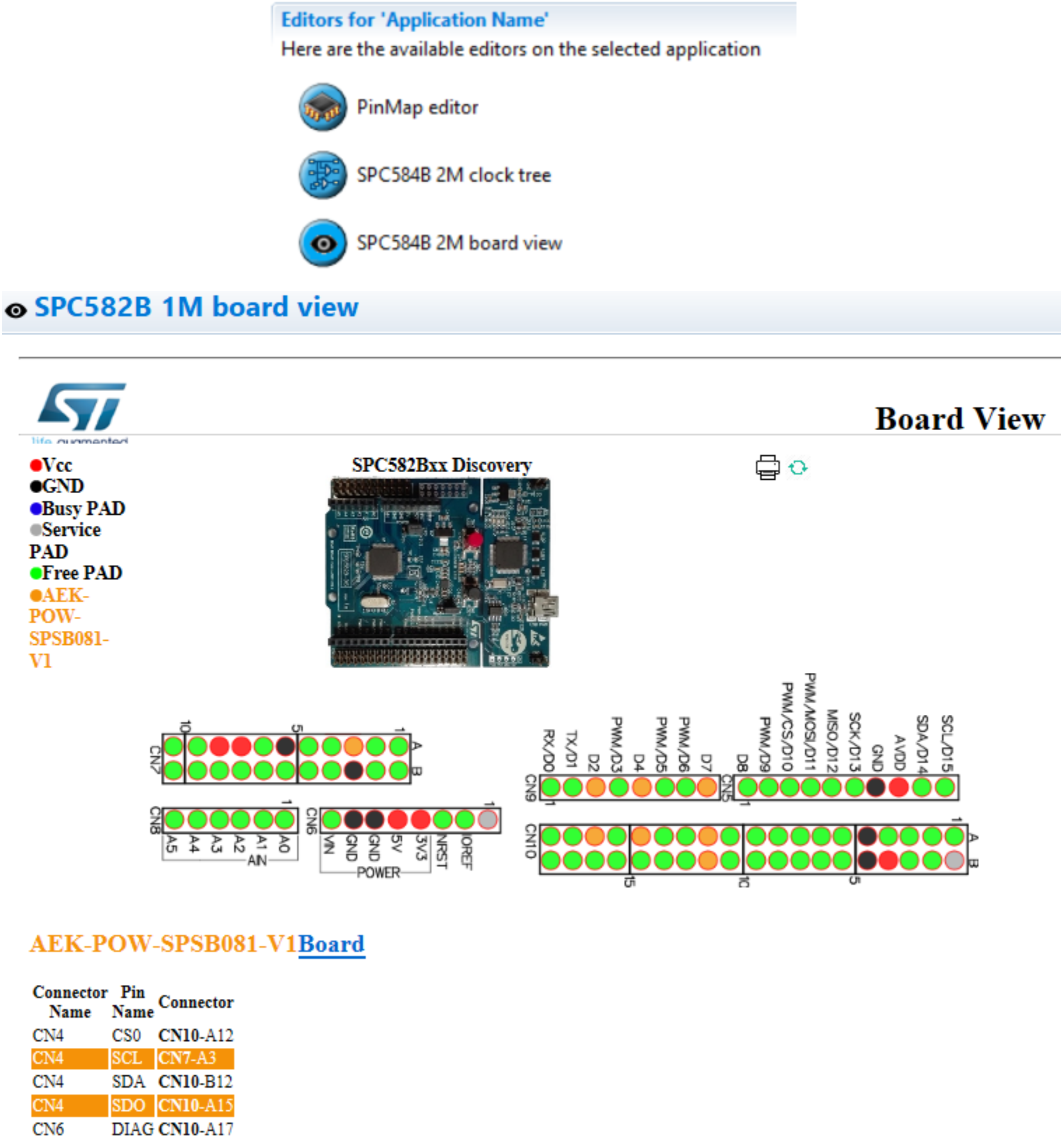
Step 7. Click on the “Allocation” button to allocate the AEK-POW-SPSB081 component.

Figure 21. Component allocation



Step 8. Click on “Board View” to view the hardware connection between the AEK-MCU-C1MLIT1 board and the AEK-POW-SPSB081.

Figure 22. Board view



Connector Name	Pin Name	Connector
CN4	CS0	CN10-A12
CN4	SCL	CN7-A3
CN4	SDA	CN10-B12
CN4	SDO	CN10-A15
CN6	DIAG	CN10-A17

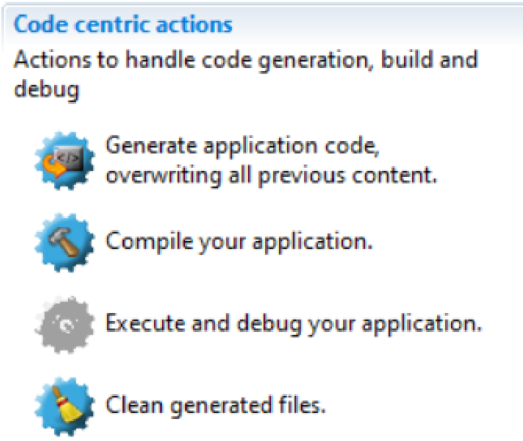
Step 9. Create your main application example as follows:

```
#include "components.h"

#include "AEK_POW_SPSB081_lld.h"

/*

* Application entry point.

*/

intmain(void) {

 componentsInit();
 irqIsrEnable();
 AEK_POW_SPSB081_init();
 /* Starting configuration */
 AEK_POW_SPSB081_WDT_Time_Config(SPSB081_DEV1,SPSB081_TSW1);
 AEK_POW_SPSB081_WU1_Input_Config(SPSB081_DEV1,SPSB081_WU1_PULLDOWN);
 AEK_POW_SPSB081_WU1_Input_Filter_Config(SPSB081_DEV1,SPSB081_WU1_FLT_STATIC);
 AEK_POW_SPSB081_Enable_Wakeup_WU1(SPSB081_DEV1);
 AEK_POW_SPSB081_Disable_Wakeup_WU2(SPSB081_DEV1);
 AEK_POW_SPSB081_DIR_WU2_Conf(SPSB081_DEV1,SPSB081_DIR_FUNCT);
 AEK_POW_SPSB081_Disable_Wakeup_CAN(SPSB081_DEV1);
 AEK_POW_SPSB081_Disable_Wakeup_LIN(SPSB081_DEV1);
 AEK_POW_SPSB081_Enable_Diag_Ext(SPSB081_DEV1);
 AEK_POW_SPSB081_Disable_V1_ICMP(SPSB081_DEV1);
 AEK_POW_SPSB081_V2_Voltage_Config(SPSB081_DEV1, SPSB081_V2_3_3_V);
 AEK_POW_SPSB081_V2_Regulator_Config(SPSB081_DEV1, SPSB081_V2_ON_IN_ACTIVE_MODE);

/* Configuring can transceiver to work in Active state*/

 AEK_POW_SPSB081_Enable_CAN_Active_mode(SPSB081_DEV1);

/* Configuration example of HS outs 1 2 3 */
AEK_POW_SPSB081_Timer1_Period_Config(SPSB081_DEV1, SPSB081_T1_PER_1000_MS);
AEK_POW_SPSB081_Timer2_Period_Config(SPSB081_DEV1, SPSB081_T2_PER_2000_MS);
AEK_POW_SPSB081_Timer1_On_Time_Config(SPSB081_DEV1, SPSB081_T1_ON_TIME_10_MS);
AEK_POW_SPSB081_Timer2_On_Time_Config(SPSB081_DEV1, SPSB081_T2_ON_TIME_20_MS);
AEK_POW_SPSB081_PWM3_duty_Conf(SPSB081_DEV1, 2);
AEK_POW_SPSB081_PWM3_Freq_Conf(SPSB081_DEV1, SPSB081_PWM3_125HZ);
AEK_POW_SPSB081_OUT1_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT1_TIMER1);
AEK_POW_SPSB081_OUT2_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT2_TIMER2);
AEK_POW_SPSB081_OUT3_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT3_PWM3);

for ( ; ; ) {

    /* Manual driving of OUT4 */
    AEK_POW_SPSB081_OUT4_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT4_ON);
    osalThreadDelayMilliseconds(500);
    AEK_POW_SPSB081_OUT4_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT4_OFF);
    osalThreadDelayMilliseconds(500);

}

}
```

Note: *This application configures the AEK-POW-SPSB081 by enabling CAN, swiching on the V2 as an LDO and enabling WU1 wakeup source.*
In this example, every high-side output is configured in four different working modes.

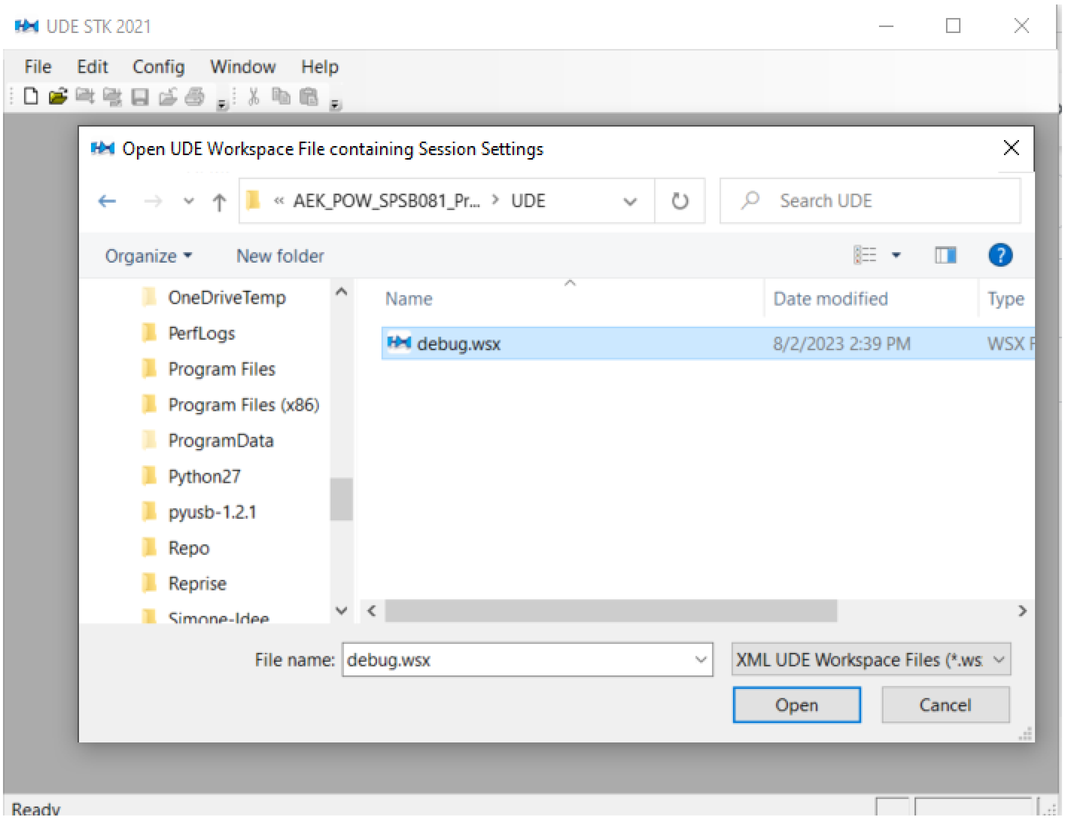
Step 10. Generate and compile your application.

Figure 23. Code generation and compilation



Step 11. Open “UDE Starterkit” and import the “.wsx” file from the workspace to flash your application.

Figure 24. Importing debug.wsx file



Step 12. Switch on your AEK-MCU-C1MLIT1 and run your AEK-POW-SPSB081 application.

5 Available demos for AEK-POW-SPSB081

In the Autodevkit release 2.2.0 (or higher), the following demos are available for the AEK-POW-SPSB081:

- *SPC582B - SPSB081 test application for discovery*, which is a demo application for the AEK-MCU-C1MLIT1 to configure an AEK-POW-SPSB081 in order to supply the AEK-MCU-C1MLIT1 and drive the four high-side outputs exploiting four different working modes;
- *SPC584B - SPSB081 test application for discovery*, which is a demo application for the *SPC584B-DIS to* configure an AEK-POW-SPSB081 in order to supply the MCU board and drive the four high-side outputs exploiting four different working modes;
- *SPC58EC - SPSB081 test application for discovery*, which is a demo application for the AEK-MCU-C4MLIT1 in order to supply the AEK-MCU-C4MLIT1 and drive the four high-side outputs exploiting four different working modes.

6 Available APIs

The APIs listed in the following tables are declared in the “AEK_POW_SPSB081_lld.h” file.

Table 7. APIs for the AEK-POW-SPSB081

API name	Description
AEK_POW_SPSB081_init(void)	Initialization function for all devices allocated. Initializes PIT for watchdog and the ADC used in the current monitoring
AEK_POW_SPSB081_Deinit(void)	Deinitialization function for all devices
AEK_POW_SPSB081_fsm(SPSB081_dev_t dev)	FSM function evaluator. Evaluates the FSM of the device, depending on the global status byte (GSB)
AEK_POW_SPSB081_write_reg(SPSB081_dev_t dev, uint8_t reg_addr, uint32_t data)	Low-level SPI write register
AEK_POW_SPSB081_readReg(SPSB081_dev_t dev, uint8_t reg_addr)	Low-level SPI read register
AEK_POW_SPSB081_readClearReg(SPSB081_dev_t dev, uint8_t reg_addr, uint32_t data)	Low-level SPI read and clear register
AEK_POW_SPSB081_readInfo(SPSB081_dev_t dev, uint8_t reg_addr)	Low-level SPI read info register
AEK_POW_SPSB081_setPIN(SPSB081_dev_t dev, uint8_t pin, uint8_t val)	Sets the logic state of the GPIO pin specified by the argument
AEK_POW_SPSB081_readPIN(SPSB081_dev_t dev, uint8_t pin)	Gets the logic state of the GPIO pin specified by the argument
AEK_POW_SPSB081_CR_dump(SPSB081_dev_t dev)	Control Register Dumping. Populates the register map related to CR registers
AEK_POW_SPSB081_SR_dump(SPSB081_dev_t dev, SPSB081_op_code_t op_code)	Status Register Dumping. Populates the register map related to SR registers.
AEK_POW_SPSB081_get_dev_sts(SPSB081_dev_t dev)	Gets SPSB0815 FSM status
AEK_POW_SPSB081_V2_Regulator_Config(SPSB081_dev_t dev, uint8_t opt)	V2 Voltage Regulator Configuration
AEK_POW_SPSB081_V2_Voltage_Config(SPSB081_dev_t dev, uint8_t opt)	V2 Voltage Configuration
AEK_POW_SPSB081_V2_Tracker_Config(SPSB081_dev_t dev, uint8_t opt)	V2 Voltage Tracker Configuration
AEK_POW_SPSB081_WDT_Time_Config(SPSB081_dev_t dev, uint8_t opt)	Watchdog Time Configuration
AEK_POW_SPSB081_V1_Reset_Config(SPSB081_dev_t dev, uint8_t opt)	V1 Reset Threshold Configuration
AEK_POW_SPSB081_Timer1_On_Time_Config(SPSB081_dev_t dev, uint8_t opt)	Timer1 TH Configuration
AEK_POW_SPSB081_Timer1_Period_Config(SPSB081_dev_t dev, uint8_t opt)	Timer1 Period Configuration
AEK_POW_SPSB081_Timer2_On_Time_Config(SPSB081_dev_t dev, uint8_t opt)	Timer2 TH Configuration
AEK_POW_SPSB081_Timer2_Period_Config(SPSB081_dev_t dev, uint8_t opt)	Timer2 Period Configuration
AEK_POW_SPSB081_WU1_Input_Config(SPSB081_dev_t dev, uint8_t opt)	Wakeup1 PullDown-Up Configuration
AEK_POW_SPSB081_WU2_Input_Config(SPSB081_dev_t dev, uint8_t opt)	Wakeup2 PullDown-Up Configuration
AEK_POW_SPSB081_WU1_Input_Filter_Config(SPSB081_dev_t dev, uint8_t opt)	Wakeup1 Input Filter Configuration
AEK_POW_SPSB081_WU2_Input_Filter_Config(SPSB081_dev_t dev, uint8_t opt)	Wakeup2 Input Filter Configuration
AEK_POW_SPSB081_Enable_Wakeup_WU1(SPSB081_dev_t dev)	Enables Wakeup1 input functionality

API name	Description
AEK_POW_SPSB081_Disable_Wakeup_WU1(SPSB081_dev_t dev)	Enables Wakeup2 input functionality
AEK_POW_SPSB081_Enable_Wakeup_WU2(SPSB081_dev_t dev)	Disables Wakeup1 input functionality
AEK_POW_SPSB081_Disable_Wakeup_WU2(SPSB081_dev_t dev)	Disables Wakeup2 input functionality
AEK_POW_SPSB081_Enable_Wakeup_LIN(SPSB081_dev_t dev)	Enables Wakeup LIN input functionality
AEK_POW_SPSB081_Disable_Wakeup_LIN(SPSB081_dev_t dev)	Disables Wakeup LIN input functionality
AEK_POW_SPSB081_Enable_Wakeup_CAN(SPSB081_dev_t dev)	Enables Wakeup CAN input functionality
AEK_POW_SPSB081_Disable_Wakeup_CAN(SPSB081_dev_t dev)	Disables Wakeup CAN input functionality
AEK_POW_SPSB081_Enable_Wakeup_Timer2(SPSB081_dev_t dev)	Enables Wakeup Timer2
AEK_POW_SPSB081_Disable_Wakeup_Timer2(SPSB081_dev_t dev)	Disables Wakeup Timer2
AEK_POW_SPSB081_Enable_Wakeup_Timer1(SPSB081_dev_t dev)	Enables Wakeup Timer1
AEK_POW_SPSB081_Disable_Wakeup_Timer1(SPSB081_dev_t dev)	Disables Wakeup Timer1
AEK_POW_SPSB081_Force_V1_StandBy(SPSB081_dev_t dev)	Forces V1 Standby FSM transition
AEK_POW_SPSB081_Force_VBAT_StandBy(SPSB081_dev_t dev)	Forces VBAT Standby FSM transition
AEK_POW_SPSB081_Force_Awake(SPSB081_dev_t dev)	Forces device Awake via SPI
AEK_POW_SPSB081_OUT1_DrvConfiguration(SPSB081_dev_t dev, uint8_t opt)	OUT1 Driver configuration
AEK_POW_SPSB081_OUT2_DrvConfiguration(SPSB081_dev_t dev, uint8_t opt)	OUT2 Driver configuration
AEK_POW_SPSB081_OUT3_DrvConfiguration(SPSB081_dev_t dev, uint8_t opt)	OUT3 Driver configuration
AEK_POW_SPSB081_OUT4_DrvConfiguration(SPSB081_dev_t dev, uint8_t opt)	OUT4 Driver configuration
AEK_POW_SPSB081_Enable_CCM_OUT1(SPSB081_dev_t dev)	Enables Constant Current Monitoring OUT1
AEK_POW_SPSB081_Enable_CCM_OUT2(SPSB081_dev_t dev)	Enables Constant Current Monitoring OUT2
AEK_POW_SPSB081_Enable_CCM_OUT3(SPSB081_dev_t dev)	Enables Constant Current Monitoring OUT3
AEK_POW_SPSB081_Enable_CCM_OUT4(SPSB081_dev_t dev)	Enables Constant Current Monitoring OUT4
AEK_POW_SPSB081_Disable_CCM_OUT1(SPSB081_dev_t dev)	Disables Constant Current Monitoring OUT1
AEK_POW_SPSB081_Disable_CCM_OUT2(SPSB081_dev_t dev)	Disables Constant Current Monitoring OUT2
AEK_POW_SPSB081_Disable_CCM_OUT3(SPSB081_dev_t dev)	Disables Constant Current Monitoring OUT3
AEK_POW_SPSB081_Disable_CCM_OUT4(SPSB081_dev_t dev)	Disables Constant Current Monitoring OUT4
AEK_POW_SPSB081_PWM1_Freq_Conf(SPSB081_dev_t dev, uint8_t opt)	PWM Frequency configuration for OUT 1
AEK_POW_SPSB081_PWM2_Freq_Conf(SPSB081_dev_t dev, uint8_t opt)	PWM Frequency configuration for OUT 2
AEK_POW_SPSB081_PWM3_Freq_Conf(SPSB081_dev_t dev, uint8_t opt)	PWM Frequency configuration for OUT 3

API name	Description
AEK_POW_SPSB081_PWM4_Freq_Conf(SPSB081_dev_t dev, uint8_t opt)	PWM Frequency configuration for OUT 4
AEK_POW_SPSB081_PWM1_duty_Conf(SPSB081_dev_t dev, float duty)	PWM duty cycle configuration for OUT 1
AEK_POW_SPSB081_PWM2_duty_Conf(SPSB081_dev_t dev, float duty)	PWM duty cycle configuration for OUT 2
AEK_POW_SPSB081_PWM3_duty_Conf(SPSB081_dev_t dev, float duty)	PWM duty cycle configuration for OUT 3
AEK_POW_SPSB081_PWM4_duty_Conf(SPSB081_dev_t dev, float duty)	PWM duty cycle configuration for OUT 4
AEK_POW_SPSB081_DIR_Conf(SPSB081_dev_t dev, uint8_t opt)	DIR functionality configuration
AEK_POW_SPSB081_Enable_CM(SPSB081_dev_t dev)	Enables Current Monitoring
AEK_POW_SPSB081_Disable_CM(SPSB081_dev_t dev)	Disables Current Monitoring
AEK_POW_SPSB081_CM_Conf(SPSB081_dev_t dev, uint8_t opt)	Current Monitoring configuration. Defines which is the OUT to be monitored
AEK_POW_SPSB081_Enable_VS_Lock(SPSB081_dev_t dev)	Enables VS Lock
AEK_POW_SPSB081_Disable_VS_Lock(SPSB081_dev_t dev)	Disables VS Lock
AEK_POW_SPSB081_Enable_VS_OV_SD(SPSB081_dev_t dev)	Enables Vs Over Voltage Shutdown
AEK_POW_SPSB081_Disable_VS_OV_SD(SPSB081_dev_t dev)	Disables Vs Over Voltage Shutdown
AEK_POW_SPSB081_Enable_VS_UV_SD(SPSB081_dev_t dev)	Enables Vs Under Voltage Shutdown
AEK_POW_SPSB081_Disable_VS_UV_SD(SPSB081_dev_t dev)	Disables Vs Under Voltage Shutdown
AEK_POW_SPSB081_Enable_VSREG_Lock(SPSB081_dev_t dev)	Enables VSREG Lock
AEK_POW_SPSB081_Disable_VSREG_Lock(SPSB081_dev_t dev)	Disables VSREG Lock
AEK_POW_SPSB081_Enable_V1_OV_Prot(SPSB081_dev_t dev)	Enables V1 OverVoltage Protection
AEK_POW_SPSB081_Disable_V1_OV_Prot(SPSB081_dev_t dev)	Disables V1 OverVoltage Protection
AEK_POW_SPSB081_Enable_ICMP_th(SPSB081_dev_t dev)	Enables ICMP Threshold
AEK_POW_SPSB081_Disable_ICMP_th(SPSB081_dev_t dev)	Disables ICMP Threshold
AEK_POW_SPSB081_Enable_V1_ICMP(SPSB081_dev_t dev)	Enables V1_ICMP
AEK_POW_SPSB081_Disable_V1_ICMP(SPSB081_dev_t dev)	Disables V1_ICMP
AEK_POW_SPSB081_Enable_IWK(SPSB081_dev_t dev)	Enables IWK generates an interrupt when waking-up from CAN or going-out from standby mode
AEK_POW_SPSB081_Disable_IWK(SPSB081_dev_t dev)	Disables IWK
AEK_POW_SPSB081_Enable_Cluster_TSD_mode(SPSB081_dev_t dev)	Enables Cluster Thermal Shutdown mode
AEK_POW_SPSB081_Disable_Cluster_TSD_mode(SPSB081_dev_t dev)	Disables Cluster Thermal Shutdown mode
AEK_POW_SPSB081_Enable_Diag_Ext(SPSB081_dev_t dev)	Enables Cluster Thermal Shutdown mode
AEK_POW_SPSB081_Disable_Diag_Ext(SPSB081_dev_t dev)	Disables Cluster Thermal Shutdown mode
AEK_POW_SPSB081_Enable_CAN_RX_only_mode(SPSB081_dev_t dev)	Enables CAN RX only mode

API name	Description
AEK_POW_SPSB081_Disable_CAN_RX_only_mode(SPSB081_dev_t dev)	Disables CAN RX only mode
AEK_POW_SPSB081_Enable_CAN_Active_mode(SPSB081_dev_t dev)	Enables CAN Active mode
AEK_POW_SPSB081_Disable_CAN_Active_mode(SPSB081_dev_t dev)	Disables CAN Active mode
AEK_POW_SPSB081_Enable_CAN_AutoBias(SPSB081_dev_t dev)	Enables CAN Autobias
AEK_POW_SPSB081_Disable_CAN_AutoBias(SPSB081_dev_t dev)	Disables CAN Autobias
AEK_POW_SPSB081_Enable_Standard_LIN_Wakeup(SPSB081_dev_t dev)	Enables Standard LIN wakeup
AEK_POW_SPSB081_Enable_Not_Standard_LIN_Wakeup(SPSB081_dev_t dev)	Enables Not Standard LIN wakeup
AEK_POW_SPSB081_get_VS_OV(SPSB081_dev_t dev)	Gets Vs OverVoltage warning
AEK_POW_SPSB081_get_VS_UV(SPSB081_dev_t dev)	Gets Vs UnderVoltage warning
AEK_POW_SPSB081_get_VSreg_OV(SPSB081_dev_t dev)	Gets VSREG OverVoltage warning
AEK_POW_SPSB081_get_VSreg_UV(SPSB081_dev_t dev)	Gets VSREG UnderVoltage warning
AEK_POW_SPSB081_get_V1_OV(SPSB081_dev_t dev)	Gets V1 OverVoltage warning
AEK_POW_SPSB081_get_V1_UV(SPSB081_dev_t dev)	Gets V1 UnderVoltage warning
AEK_POW_SPSB081_get_V1_Fail(SPSB081_dev_t dev)	Gets V1 Fail
AEK_POW_SPSB081_get_V2_Fail(SPSB081_dev_t dev)	Gets V2 Fail
AEK_POW_SPSB081_get_V2_SC(SPSB081_dev_t dev)	Gets V2 Short Circuit
AEK_POW_SPSB081_get_TW(SPSB081_dev_t dev)	Gets Thermal Warning
AEK_POW_SPSB081_get_TSHDN1(SPSB081_dev_t dev)	Gets Thermal Shutdown1
AEK_POW_SPSB081_get_TSHDN2(SPSB081_dev_t dev)	Gets Thermal Shutdown2
AEK_POW_SPSB081_get_LIN_TXD_Dom(SPSB081_dev_t dev)	Gets LIN TXD Dominant
AEK_POW_SPSB081_get_LIN_PERM_Dom(SPSB081_dev_t dev)	Gets LIN PERM Dominant
AEK_POW_SPSB081_get_LIN_PERM_Rec(SPSB081_dev_t dev)	Gets LIN PERM Recessive
AEK_POW_SPSB081_get_CAN_TXD_Dom(SPSB081_dev_t dev)	Gets CAN TXD Dominant
AEK_POW_SPSB081_get_CAN_RXD_Rec(SPSB081_dev_t dev)	Gets CAN TXD Recessive
AEK_POW_SPSB081_get_CAN_PERM_Dom(SPSB081_dev_t dev)	Gets CAN PERM Dominant
AEK_POW_SPSB081_get_CAN_PERM_Rec(SPSB081_dev_t dev)	Gets CAN PERM Recessive
AEK_POW_SPSB081_get_CAN_TO(SPSB081_dev_t dev)	Gets CAN Timeout
AEK_POW_SPSB081_get_SPI_INV_Cmd(SPSB081_dev_t dev)	Gets SPI Invalid Command
AEK_POW_SPSB081_get_SPI_CLK_Count(SPSB081_dev_t dev)	Gets SPI Clock Count
AEK_POW_SPSB081_get_WD_Fail(SPSB081_dev_t dev)	Gets Watchdog Failure
AEK_POW_SPSB081_get_Forced_Sleep_TSD2(SPSB081_dev_t dev)	Gets Forced Sleep Thermal Shutdown2
AEK_POW_SPSB081_get_Forced_Sleep_V1OV(SPSB081_dev_t dev)	Gets Forced Sleep V1 OverVoltage
AEK_POW_SPSB081_get_Forced_Sleep_WD(SPSB081_dev_t dev)	Gets Forced Sleep Watchdog
AEK_POW_SPSB081_get_POR(SPSB081_dev_t dev)	Gets Power On Reset
AEK_POW_SPSB081_get_WD_Fail_Cnt(SPSB081_dev_t dev)	Gets Watchdog failure count
AEK_POW_SPSB081_get_V1_Restart_Cnt(SPSB081_dev_t dev)	Gets V1 Restart Count
AEK_POW_SPSB081_get_CAN_SUP_Low(SPSB081_dev_t dev)	Gets CAN Supply Low
AEK_POW_SPSB081_get_IP_SUP_Low(SPSB081_dev_t dev)	Gets IP Supply Low
AEK_POW_SPSB081_get_SGNDLOSS(SPSB081_dev_t dev)	Gets SGND Loss
AEK_POW_SPSB081_get_OUT1_OC(SPSB081_dev_t dev)	Gets OUT1 Over Current

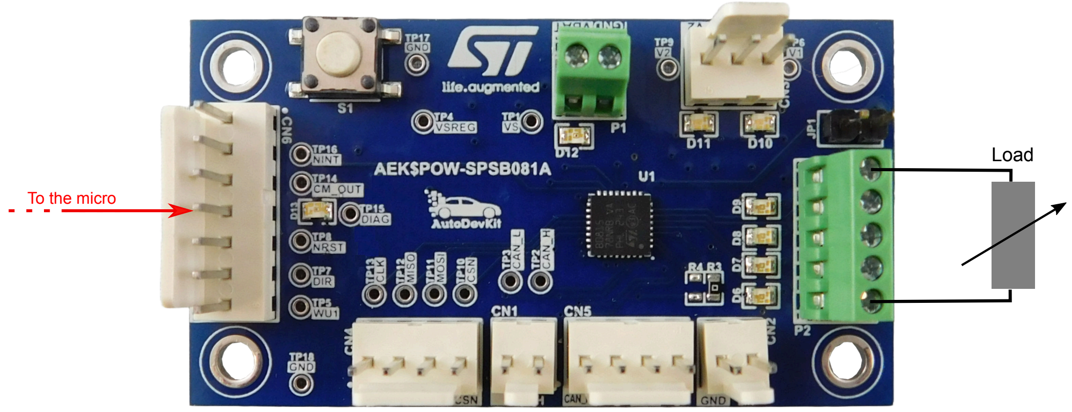
API name	Description
AEK_POW_SPSB081_get_OUT2_OC(SPSB081_dev_t dev)	Gets OUT2 Over Current
AEK_POW_SPSB081_get_OUT3_OC(SPSB081_dev_t dev)	Gets OUT3 Over Current
AEK_POW_SPSB081_get_OUT4_OC(SPSB081_dev_t dev)	Gets OUT4 Over Current
AEK_POW_SPSB081_get_OUT1_OL(SPSB081_dev_t dev)	Gets OUT1 Open Load
AEK_POW_SPSB081_get_OUT2_OL(SPSB081_dev_t dev)	Gets OUT2 Open Load
AEK_POW_SPSB081_get_OUT3_OL(SPSB081_dev_t dev)	Gets OUT3 Open Load
AEK_POW_SPSB081_get_OUT4_OL(SPSB081_dev_t dev)	Gets OUT4 Open Load
AEK_POW_SPSB081_get_TW1(SPSB081_dev_t dev)	Gets Thermal Warning1
AEK_POW_SPSB081_get_TW2(SPSB081_dev_t dev)	Gets Thermal Warning2
AEK_POW_SPSB081_get_TW3(SPSB081_dev_t dev)	Gets Thermal Warning3
AEK_POW_SPSB081_get_TW4(SPSB081_dev_t dev)	Gets Thermal Warning4
AEK_POW_SPSB081_get_TSD1(SPSB081_dev_t dev)	Gets Thermal Shutdown1
AEK_POW_SPSB081_get_TSD2(SPSB081_dev_t dev)	Gets Thermal Shutdown2
AEK_POW_SPSB081_get_TSD3(SPSB081_dev_t dev)	Gets Thermal Shutdown3
AEK_POW_SPSB081_get_TSD4(SPSB081_dev_t dev)	Gets Thermal Shutdown4.
AEK_POW_SPSB081_get_NINT(SPSB081_dev_t dev)	Gets NINT Pin level
AEK_POW_SPSB081_get_DIAG(SPSB081_dev_t dev)	Gets DIAG Pin level
AEK_POW_SPSB081_get_NRST(SPSB081_dev_t dev)	Gets NRST Pin level
AEK_POW_SPSB081_get_DIR(SPSB081_dev_t dev)	Gets DIR Pin level
AEK_POW_SPSB081_get_WU1(SPSB081_dev_t dev)	Gets WU1 Pin level
AEK_POW_SPSB081_get_CM(SPSB081_dev_t dev)	Gets Current Monitoring Level
AEK_POW_SPSB081_CR_setting(SPSB081_dev_t dev)	Gets OUT1 Open Load

7 Waveforms

7.1 Current monitoring use-case

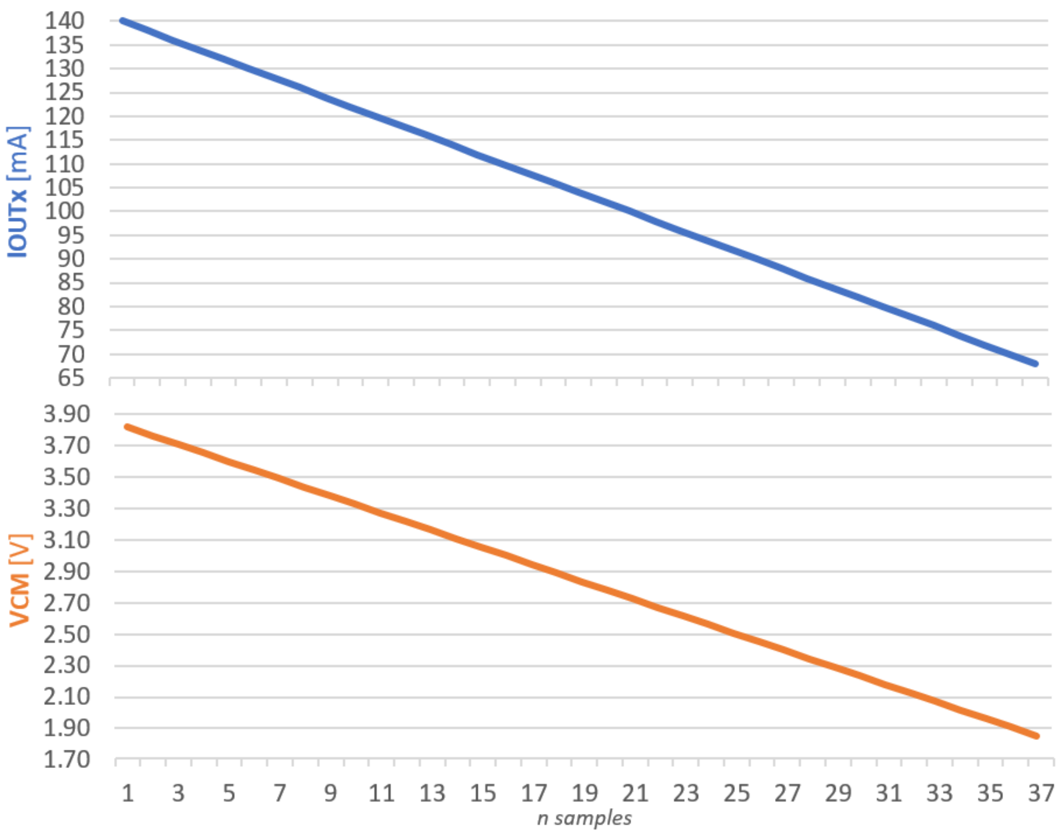
In this test, the SPSB081 has been configured in order to monitor the current absorbed by OUT1.

Figure 25. **Current monitoring use-case**



It has been performed a series of IOUT, VCM measurements. By varying the resistance of the LOAD from 80 to 200 Ω, the Current Monitoring feature has been validated.

Figure 26. **Current monitoring test results**



As shown by the VCM values, by connecting an active load to OUT1, when the resistor used as load increases, the current measured by the current monitor pin decreases according to the above formula:
VCM=(Iout1/990)*Rsense

Considering that for the AEK-POW-SPSB081:

- Rsense = 27 KΩ
- VCM max = 4 V
- Ioutx max =140 mA

7.2 HS output use-case example

To test the high-side outputs, we used the following code:

```
//setting duty conf out
AEK_POW_SPSB081_PWM2_duty_Conf(SPSB081_DEV1, 2); AEK_POW_SPSB081_PWM2_Freq_Conf(SPSB081_DEV1
, SPSB081_PWM2_250HZ);
//setting timers
AEK_POW_SPSB081_Timer1_Period_Config(SPSB081_DEV1, SPSB081_T1_PER_100_MS);
AEK_POW_SPSB081_Timer2_Period_Config(SPSB081_DEV1, SPSB081_T2_PER_2000_MS);
AEK_POW_SPSB081_Timer1_On_Time_Config(SPSB081_DEV1, SPSB081_T1_ON_TIME_10_MS);
AEK_POW_SPSB081_Timer2_On_Time_Config(SPSB081_DEV1, SPSB081_T2_ON_TIME_20_MS);
//setting modes for every out
AEK_POW_SPSB081_OUT1_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT1_ON);
AEK_POW_SPSB081_OUT2_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT2_PWM2);
AEK_POW_SPSB081_OUT3_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT3_TIMER1);
AEK_POW_SPSB081_OUT4_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT4_TIMER2);
//out1 on/off in loop every 200ms
 for ( ; ; ) {
AEK_POW_SPSB081_OUT1_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT1_ON);
osalThreadDelayMilliseconds(200);
AEK_POW_SPSB081_OUT1_DrvConfiguration(SPSB081_DEV1, SPSB081_OUT1_OFF);
osalThreadDelayMilliseconds(200);
}
```

Here the 4 outputs are configured using AutoDevKit driver APIs in 4 different modes:

OUT1 is switched on and off manually, every 0.2 seconds

OUT2 is configured to supply a 250 hz PWM, with a 2% duty cycle

OUT3 is configured in Timed mode, using Timer 1 settings (period 100 ms, on-time 10 ms)

OUT4 is configured in Timed mode, using Timer 2 settings (period 2000 ms, on-time 20 ms)

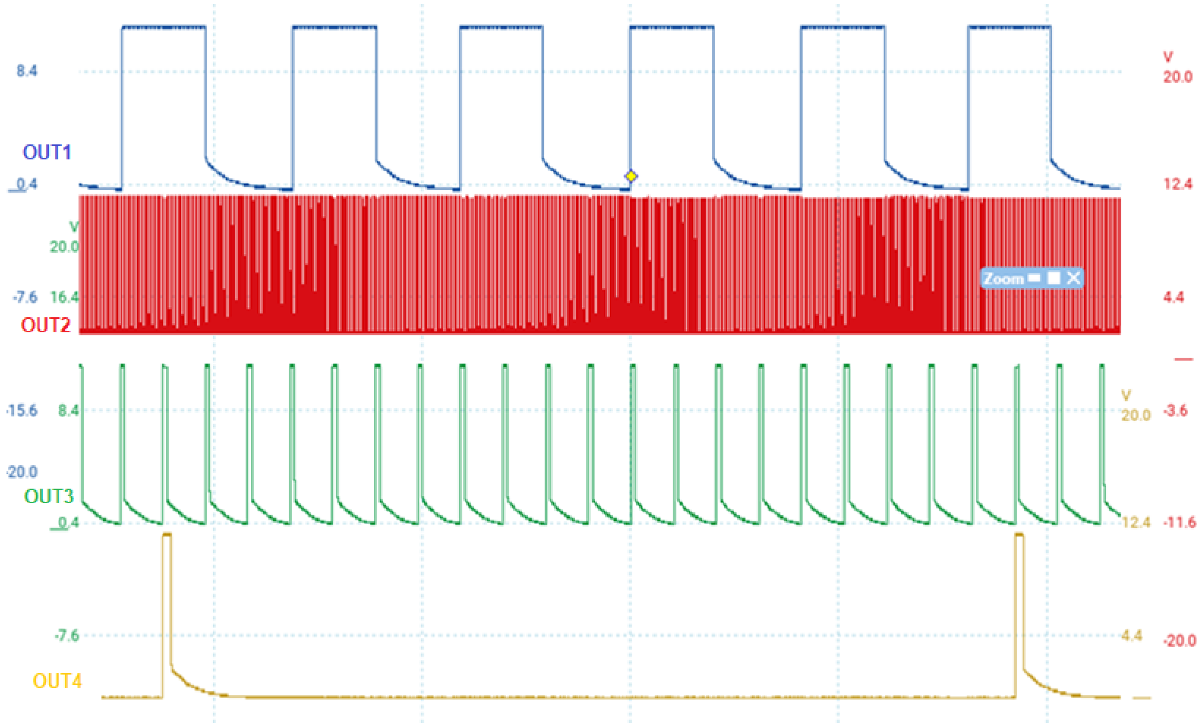


Figure 27. High-side outputs - no load

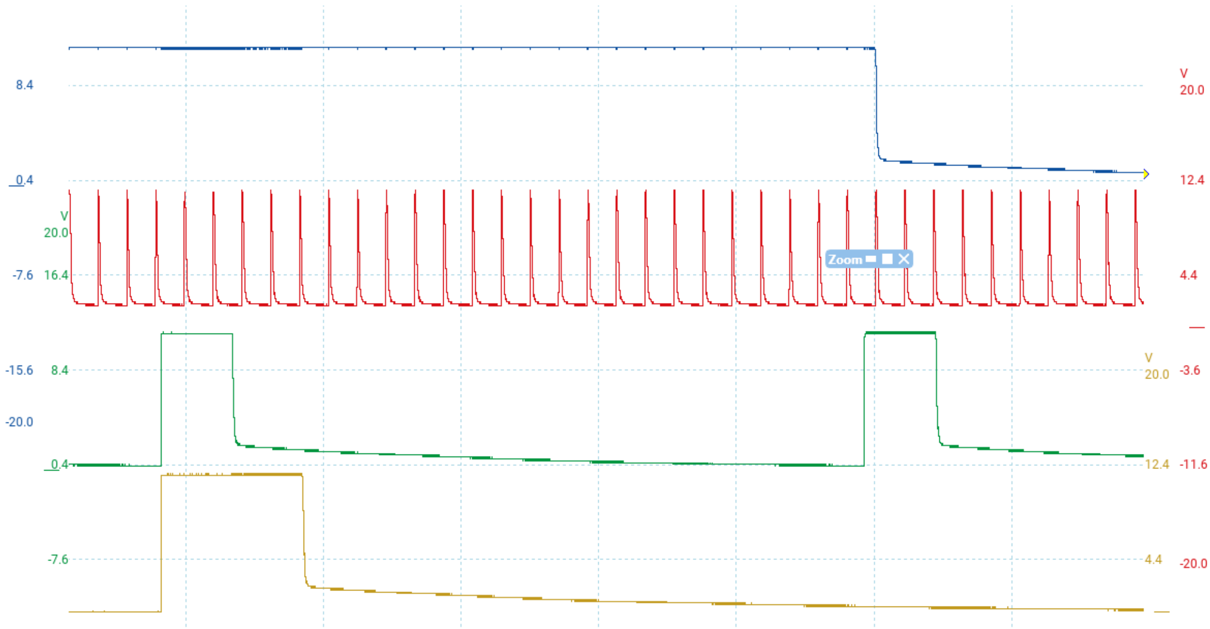


Figure 28. High-side outputs - no load (zoom)

7.3 V1/V2 test

V1 regulator supplies 5 V, whereas V2 regulator can be configured as tracker of V1 or as an LDO, supplying 3.3 V or 5 V.

To test the voltage regulators, we used the following code:

```
AEK_POW_SPSB081_V2_Tracker_Config(SPSB081_DEV1,SPSB081_V2_LINEAR_INDIP_LDO);
AEK_POW_SPSB081_V2_Regulator_Config(SPSB081_DEV1, SPSB081_V2_ON_IN_ACTIVE_MODE);
AEK_POW_SPSB081_V2_Voltage_Config(SPSB081_DEV1, SPSB081_V2_5_V); osalThreadDelayMillisecond
s(500);
AEK_POW_SPSB081_V2_Voltage_Config(SPSB081_DEV1, SPSB081_V2_3_3_V); osalThreadDelayMillisecon
ds(500);
AEK_POW_SPSB081_V2_Regulator_Config(SPSB081_DEV1, SPSB081_V2_OFF);
osalThreadDelayMilliseconds(250);
AEK_POW_SPSB081_V2_Tracker_Config( SPSB081_DEV1,SPSB081_V2_TKR_V1_REG);
AEK_POW_SPSB081_V2_Regulator_Config(SPSB081_DEV1, SPSB081_V2_ON_IN_ACTIVE_MODE);
osalThreadDelayMilliseconds(400);
AEK_POW_SPSB081_Force_VBAT_StandBy(SPSB081_DEV1);
```

Here the V2 regulator is configured as follows:

A: V2 ON_during_Active_mode, 5 V LDO

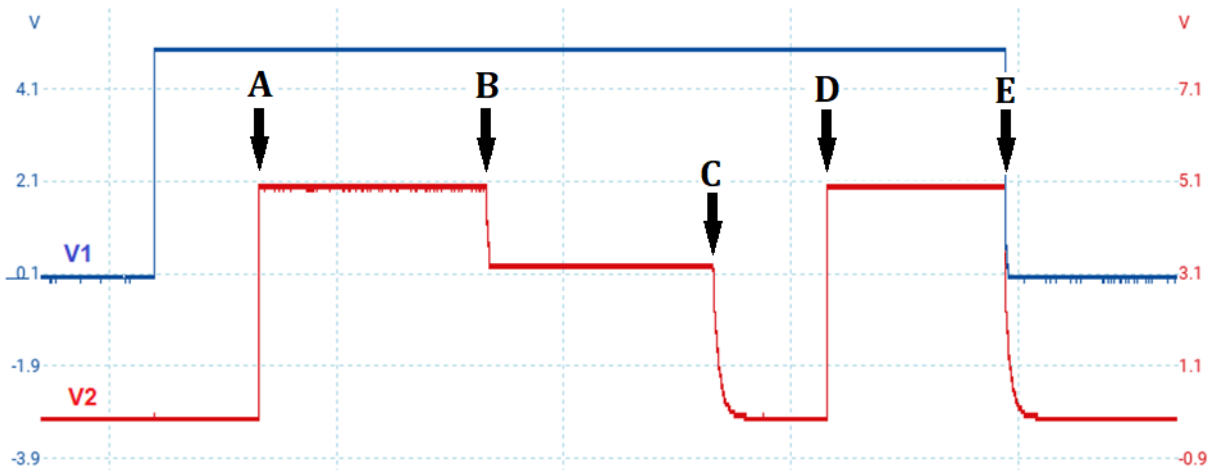
B: V2 3.3 V LDO

C: V2 switched OFF

D: V2 ON_during_Active_mode, Voltage Tracker of V1

E: VBAT standby → V1 and V2 switched OFF

Figure 29. **V1 and V2 regulators test waveforms**

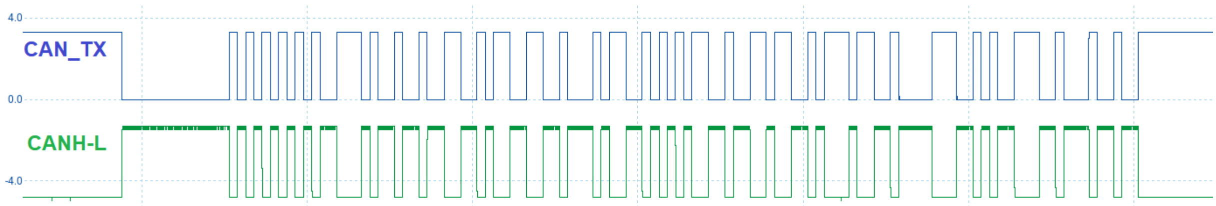


7.4 CAN test

This test has been performed in active mode, using the following code string:
AEK_POW_SPSB081_Enable_CAN_Active_mode(SPSB081_DEV1);

By configuring the CAN2 line of the microntroller board, and by setting the CAN_TX pin as very strong drive, we periodically transmitted a CAN frame (0xDDEEFFAAUL) to the AEK-POW-SPSB081 CAN_TX pin.

Figure 30. **CAN test waveforms**



Monitoring the output signal between CAN-H/CAN-L, we notice that the message is correctly transmitted through the AEK-POW-SPSB081.

7.5 CAN wake-up test

By launching this API function:

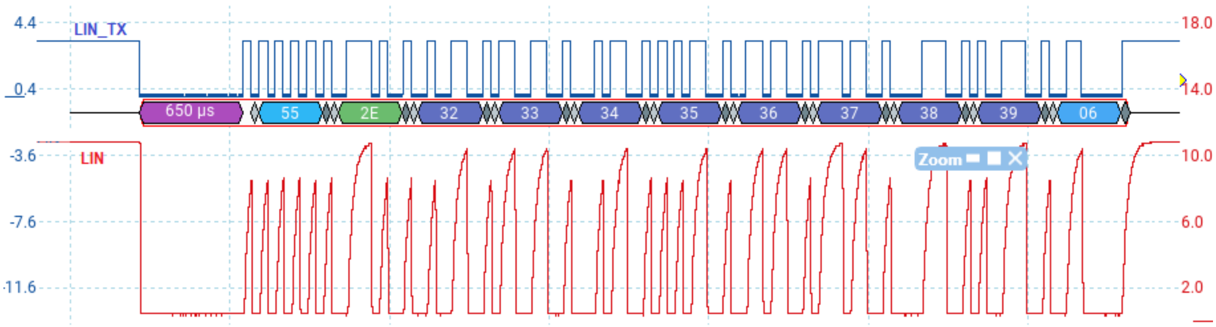
```
AEK_POW_SPSB081_Enable_Wakeup_CAN(SPSB081_DEV1);
```

the wake-up via CAN will be enabled. In this way, the board will wake up as soon as it receives a CAN message.

7.6 LIN test

By configuring a LIN line of the microcontroller and setting the LIN_TX pin as very strong drive, we transmitted a LIN frame (0xDDEEFFAAUL) to the AEK-POW-SPSB081 LIN_TX pin.

Figure 31. LIN test waveform



The above waveform shows that the LIN pin mirrors the message received from the LIN_Tx pin.

7.7 LIN wake-up test

Enable wake-up through the following API call:

AEK_POW_SPSB081_Enable_Wakeup_LIN(SPSB081_DEV1);

Defining also the LIN message behavior, we want to define as a wake-up event one of the following functions:

```
AEK_POW_SPSB081_Enable_Not_Standard_LIN_Wakeup(SPSB081_DEV1);
```

or

```
AEK_POW_SPSB081_Enable_Standard_LIN_Wakeup(SPSB081_DEV1);
```

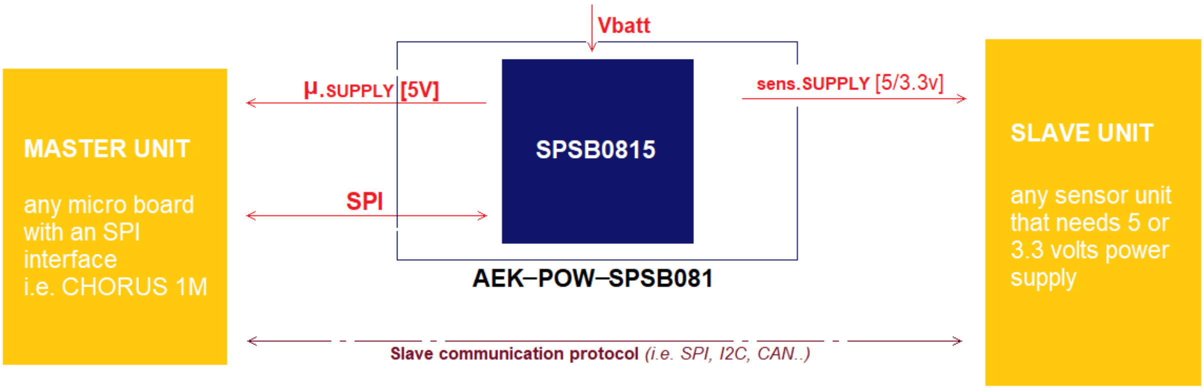
And, by sending a LIN message along the LIN line, the AEK-POW-SPSB081 wakes up.

8 Thermal behavior

The AEK-POW-SPSB081 has four layers. Its dimensions are 60x33 mm with a thickness of 1.6 mm.

Through a thermal camera, we analyzed the SPSB081 heating under some standard board operating conditions. The device heats, but its temperature never goes over the device thermal limits.

Figure 32. Thermal behavior



The case studied was the “Example Application n2” (see section 3.1), in which the AEK-POW-SPSB081 supplies a 5 V microcontroller and a 5 V sensor.

Scenario 0: (V1 ON, V2 OFF). V1 absorbing 150mA. V2 absorbing 0mA, -> Temperature=76°

Scenario 1: (V1 ON, V2 ON). V1 absorbing 150mA, V2 absorbing 50mA, -> Temperature=89°.

Scenario 2: (V1 ON, V2 ON). V1 absorbing 150mA, V2 absorbing 100mA, -> Temperature=101°

Scenario 3: (V1 ON, V2 ON). V1 absorbing 150mA, V2 absorbing 150mA, -> Temperature=111°

Figure 33. Scenario 0 on the left, Scenario 1 on the right

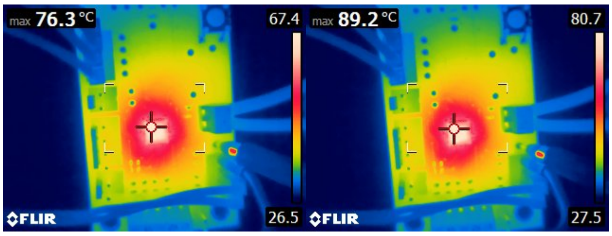
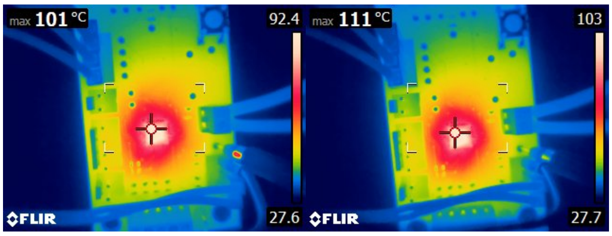
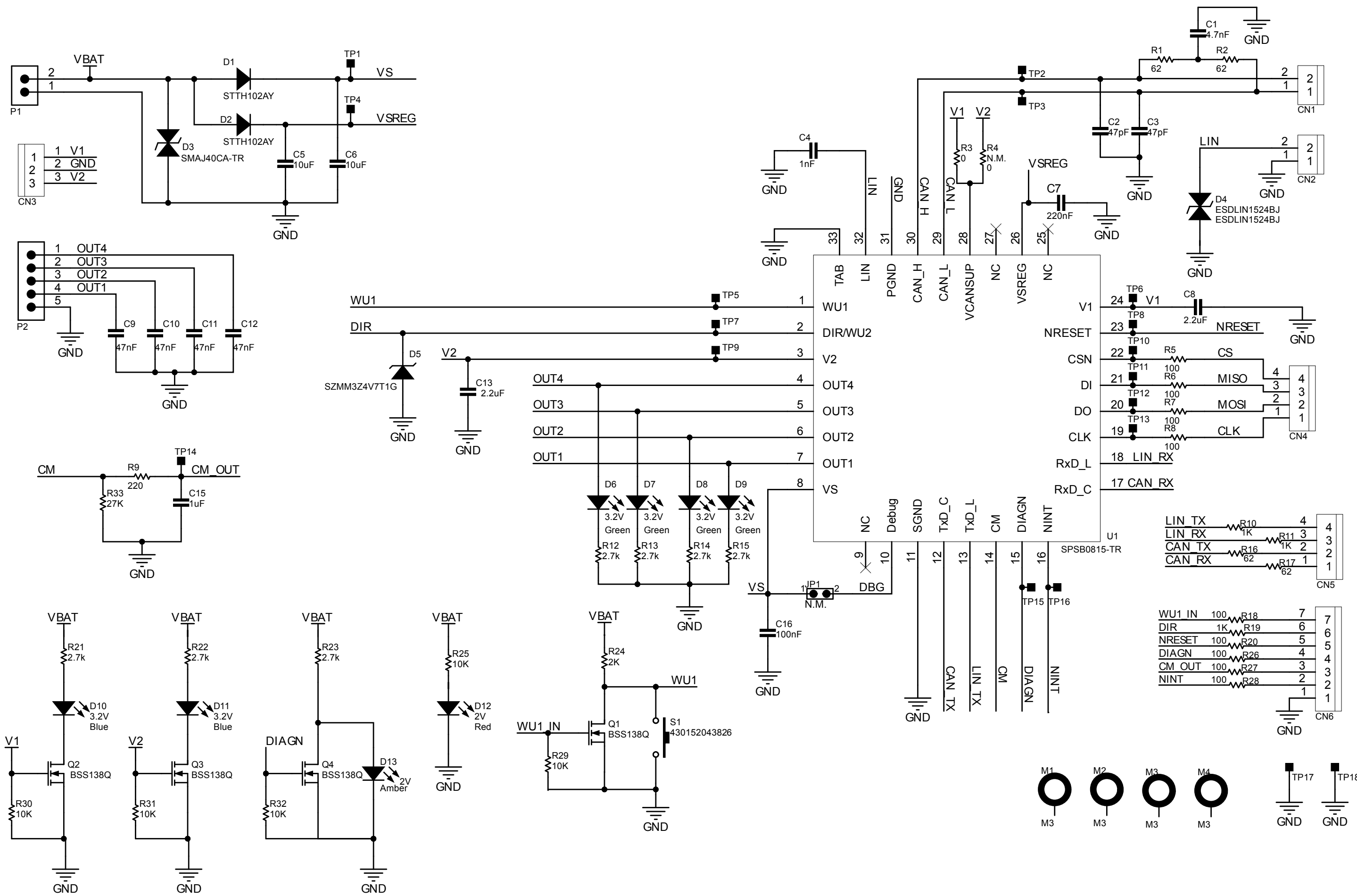


Figure 34. Scenario 2 on the left, Scenario 3 on the right



9 Schematic diagrams

Figure 35. **AEK-POW-SPSB081 circuit schematic**



10 Bill of materials

Table 8. AEK-POW-SPSB081 bill of materials

Item	Q.ty	Ref.	Part/value	Description	Manufacturer	Order code
1	1	C1	4.7nF 0603C 50V ±10%	0603 - 50V - X7R Class II	WE	885012206087
2	2	C2, C3	47pF 0603C 50V ±5%	0603 - 50V - NP0 Class I	WE	885012006055
3	1	C4	1nF 0603C 50V ±10%	0603 - 50V - X7R Class II	WE	885012206083
4	2	C5, C6	10uF 1210C 50V ±10%	1210 - 50V - X7R Class II	WE	885012209073
5	1	C7	220nF 0603C 50V ±5%	0603 - 50V - X7R Class II	WE	885012206125
6	2	C8, C13	2.2uF 0805C 25V ±10%	0805 - 25V - X7R Class II	WE	885012207079
7	4	C9, C10, C11, C12	47nF 0603C 50V ±10%	0603 - 50V - X7R Class II	WE	885012206093
8	1	C15	1uF 0805C 50V ±10%	0805 - 50V - X7R Class II	WE	885012207103
9	1	C16	100nF 0805C 100V ±10%	0805 - 100V - X7R Class II	WE	885012207128
10	2	CN1, CN2	KK254_2p 250VAC 3A	2.54mm - 1 row - KK254 - Male	WE	61900211121
11	1	CN3	KK254_3p 250VAC 3A	2.54mm - 1 row - KK254 - Male	WE	61900311121
12	2	CN4, CN5	KK254_4p 250VAC 3A	2.54mm - 1 row - KK254 - Male	WE	61900411121
13	1	CN6	KK254_7P 250VAC 3A	2.54mm - 1 row - KK254 - Male	WE	61900711121
14	2	D1, D2	STTH102AY SMA 200V 1A	Automotive high efficiency ultrafast diode.	ST	STTH102AY
15	1	D3	SMAJ40CA-TR SMA 40V 400W	Automotive 600 W, 30.8 V TVS in SMB	ST	SMAJ40CA-TR
16	1	D4	ESDLIN1524BJ SOD-323	Automotive Transil™, transient voltage suppressor (TVS) for LIN bus	ST	ESDLIN1524BJ
17	1	D5	SZMM3Z4V7T1G SOD-323 4.7V 300mW	4.7V Zener Voltage Regulators, 300mW	Onsemi	SZMM3Z4V7T1G
18	4	D6, D7, D8, D9	Green LED_0805 3.2V	0805 - Led Green - 3.2V	WE	150080GS75000
19	2	D10, D11	Blue LED_0805 3.2V	0805 - Led Blue - 3.2V	WE	150080BS75000
20	1	D12	Red LED_0805 2V	0805 - Led Red - 2V	WE	150080RS75000

Item	Q.ty	Ref.	Part/value	Description	Manufacturer	Order code
21	1	D13	Amber LED_0805 2V	0805 - Led Amber - 2V	WE	150080AS75000
22	1	JP1	SIL_2x1 250VAC 3A	2.54mm - 1 row	WE	61300211121
23	1	P1	Con_2P_2.54 150V (AC); 450 V (AC) 6A	2.54mm - WR-TBL Series 2109 - Horizontal Entry Modular	WE	691210910002
24	1	P2	Con_5P_2.54 150V (AC); 450 V (AC) 6A	2.54mm - WR-TBL Series 2109 - Horizontal Entry Modular	WE	691210910005
25	4	Q1, Q2, Q3, Q4	BSS138Q SOT-23 50 0.2A 0.3W	N-Channel Enhancement Mosfet	NEXPERIA	BSS138Q-7-F
26	4	R1, R2, R16, R17	62 0603R 0.1W ±0.1%	0603 - ±0.1% - 0.1W	Panasonic	ERA3AEB620V
27	1	R3	0 0603R 0.1W ±1%	0603 - ±1% - 0.1W	Panasonic	ERJ3GEY0R00V
28	1	R4	N.M. 0603R	0603	N.A.	N.A.
29	9	R5, R6, R7, R8, R18, R20, R26, R27, R28	100 0603R 0.25W ±1%	0603 - ±1% - 0.25W	Panasonic	ERJPA3F1000V
30	1	R9	220 0603R 0.25W ±1%	0603 - ±1% - 0.25W	Panasonic	ERJPA3F2200V
31	3	R10, R11, R19	1K 0603R 0.25W ±1%	0603 - ±1% - 0.25W	Panasonic	ERJPA3F1001V
32	7	R12, R13, R14, R15, R21, R22, R23	2.7k 0603R 0.125W ±1%	0603 - ±1% - 0.125W	Vishay	MCT06030C2701FP500
33	1	R24	2K 0603R 0.2W ±1%	0603 - ±1% - 0.2W	Panasonic	ERJP03F2001V
34	5	R25, R29, R30, R31, R32	10K 0603R 0.2W ±1%	0603 - ±1% - 0.2W	Panasonic	ERJP03F1002V
35	1	R33	27K 0805R 0.125W	0805 - ±1% - 0.125W	Panasonic	ERJ6ENF2702V
36	1	S1	430152043826 430152043826	Switch	WE	430152043826
37	1	U1	SPSB0815-TR QFN32L	Automotive Power management IC with LIN and CAN-FD	ST	SPSB0815-TR
38	4	for blister	97790403111 Screw	WA-SCRW Pan Head Screw w. cross slot M3	WE	97790403111
39	1	for blister	60900213421 Jumper	WR-PHD 2.54 mm Multi-Jumper Jumper with Test Point	WE	60900213421

Item	Q.ty	Ref.	Part/value	Description	Manufacturer	Order code
40	2	for blister	61900411621 Terminal Housing	WR-WTB 2.54 mm Female Terminal Housing	WE	61900411621
41	2	for blister	61900211621 Terminal Housing	WR-WTB 2.54 mm Female Terminal Housing	WE	61900211621
42	1	for blister	61900311621 Terminal Housing	WR-WTB 2.54 mm Female Terminal Housing	WE	61900311621
43	1	for blister	61900711621 Terminal Housing	WR-WTB 2.54 mm Female Terminal Housing	WE	61900711621
44	22	for blister	61910113722 Crimp Contact	WR-WTB 2.54 mm Female Crimp Contact	WE	61910113722
45	4	for blister	970080365 Spacer M3x8	WA-SPAII Plastic Spacer Stud, metric, internal/ internal	WE	970080365

11 Board versions

Table 9. AEK-POW-SPSB081 versions

PCB version	Schematic diagrams	Bill of materials
AEK$POW-SPSB081A [1]	AEK$POW-SPSB081A schematic diagrams	AEK$POW-SPSB081A bill of materials

1. *This code identifies the AEK-POW-SPSB081 evaluation board first version. It is printed on the board PCB.*

12 Regulatory compliance information

Notice for US Federal Communication Commission (FCC)

For evaluation only; not FCC approved for resale

FCC NOTICE - This kit is designed to allow:

(1) Product developers to evaluate electronic components, circuitry, or software associated with the kit to determine

whether to incorporate such items in a finished product and

(2) Software developers to write software applications for use with the end product.

This kit is not a finished product and when assembled may not be resold or otherwise marketed unless all required FCC equipment authorizations are first obtained. Operation is subject to the condition that this product not cause harmful interference to licensed radio stations and that this product accept harmful interference. Unless the assembled kit is designed to operate under part 15, part 18 or part 95 of this chapter, the operator of the kit must operate under the authority of an FCC license holder or must secure an experimental authorization under part 5 of this chapter 3.1.2.

Notice for Innovation, Science and Economic Development Canada (ISED)

For evaluation purposes only. This kit generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to Industry Canada (IC) rules.

À des fins d'évaluation uniquement. Ce kit génère, utilise et peut émettre de l'énergie radiofréquence et n'a pas été testé pour sa conformité aux limites des appareils informatiques conformément aux règles d'Industrie Canada (IC).

Notice for the European Union

This device is in conformity with the essential requirements of the Directive 2014/30/EU (EMC) and of the Directive 2015/863/EU (RoHS).

Notice for the United Kingdom

This device is in compliance with the UK Electromagnetic Compatibility Regulations 2016 (UK S.I. 2016 No. 1091) and with the Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment Regulations 2012 (UK S.I. 2012 No. 3032).

Revision history

Table 10. Document revision history

Date	Revision	Changes
22-Nov-2023	1	Initial release.
23-Apr-2024	2	Updated Section 1.1: Board main components.

Contents

List of tables

List of figures