AH1002

Application Hints

Core System Basis Chip UJA107xA

Version 1.2 — 17 March 2016

Document information

Info	Content
Title	**UJA107xA Core System Basis Chip Application Hints**
Author(s)	Jens Fleischhacker, Janett Habermann, Peter Buehring
Department	Systems & Applications
Keywords	UJA107xA, UJA1075A, UJA1076A, UJA1078A, UJA1079A, System Basis Chip

Summary

The intention of this document is to provide the necessary information for hardware and software designers for creation of automotive applications based on the UJA107xA Core System Basis Chip family.

For ease of description, the explanations of the hardware design and of the software flow use the UJA1078A as reference device, in which all CAN and LIN transceivers are available. For the family members UJA1075A/76A/79A the descriptions referring to CAN/LIN transceiver(s) that are not available in those derivatives have to be ignored.

Contact information

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

Revision history

Rev	Date	Description
1.2	2016-03-17	- Section 2.1: additional info about connection of TEST pins added. - Section 2.2.1: additional info about operation at supply voltage > 28V. Fig 4 updated. Remark about usage of suppressor diodes added. - New section 2.2.3 - Section 2.3.1.1: Fig 11 and its description updated - New section 2.3.4 - Section 2.4.1: more explanation and info added about usage of ceramic capacitors and about quiescent current with V1 output current > 0. Section split up into sub-sections. Subsection 2.4.1.4 added. - Section 2.4.2 revised, Table 6 updated, footnote added to table. Fig 21: capacitor added to emitter of PNP transistor. - Section 2.4.3.1: recommended minimum capacitance changed. - Section 2.7.1: Added remark about external circuitry connected to WAKE pins - Section 2.9.3: Paragraph appended about setting WMC = 1 - Section 2.13: Fig 47: pin names corrected. Optional TEST1/2 pin connection to GND via resistor added. - Section 3.3.2: reference to bit names CWIE and LWIE removed - New section 3.3.4 - Section 4 updated - Old section 5.1 about in-pdf attachments removed - Section 5.3: formulas for LIN1 and LIN2 updated. - Reference 2 updated - Editorial changes

Contact information

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

1.1	2011/01/12	- Fig 1: Added pin numbers to pinning - Section 2.4.1: Information about ESR of capacitor for V1 stability added - Section 2.4.2: Remark added that VEXCTRL cannot directly be used to control MOSFETs - Table 6: PNP power dissipation and transistor type in last line corrected and example below the table updated - Section 2.4.3.1: Information about ESR of capacitor for V2 stability added - Section 2.6.4, Fig 29, Fig 30: Statement about external termination resistor for LIN slave applications added - Section 2.7.1: Remark added how to handle not used WAKE pins - Section 2.7.3: Remark added about ESD robustness according to IEC 61000-4-2 - Fig 32, Fig 33: Added 1k series resistor for 6kV ESD robustness according to IEC 61000-4-2 - Table 10, section 2.8.2.1, section 2.8.2.2: Condition for support of V1 and RSTN clamped low detection added - Section 2.9.5: Characteristics for UJA107xA types with hardware deactivated watchdog - Section 3.2.3: Additional information on CAN Transceiver and CAN PE activation sequence added - Section 5.3 added: Explanation of power dissipation calculator - Updated attachment: Power dissipation calculator - Corrected typing errors
1.0	2010/06/25	Initial version

Contact information

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

Contents

continued >>

1. Introduction

The UJA107xA Core System Basis Chip (SBC) replaces several basic components, which are common in any automotive Electronic Control Unit (ECU), with a high-speed Controller Area Network (CAN) and two Local Interconnect Network (LIN) interfaces.

The UJA107xA supports all networking applications that control various power and sensor peripherals by using high-speed CAN as the main network interface and LIN as local sub-busses. Its package and pinning allow easy switching within the UJA107xA Core SBC family, or migration from/to the UJA106x SBC family members.

The UJA107xA is a family of automotive System Basis Chips (SBC), each mainly consisting of

- 3.3 V or 5 V voltage regulator V1 for supply of a microcontroller, up to 250 mA
- Second 5 V voltage regulator V2, e.g. for supply of the internal CAN Transceiver
- Window watchdog with Limp-Home mode control and Cyclic Wake options
- Wake input pins with support of cyclic sense
- High-speed CAN and/or LIN Transceivers.

The main type number indicates the available internal transceivers (Table 1).

Table 1. Main type numbers of the UJA107xA SBC family

Features	UJA1075A	UJA1076A	UJA1078A	UJA1079A
High-Speed CAN	Yes	Yes	Yes	No
LIN	Yes, 1x	No	Yes, 2x	Yes, 1x

For each main type number four sub-types exist (Table 2).

Table 2. Sub-types of the UJA107xA SBC family

Sub-type	V1 voltage	Watchdog
/5V0/WD	5 V	Yes
/3V3/WD	3.3 V	Yes
/5V0	5 V	No
/3V3	3.3 V	No

For more information please refer to the data sheet.

Due to the family approach and for PCB reuse, all SBCs are available in the same 32-pin HTSSOP package.

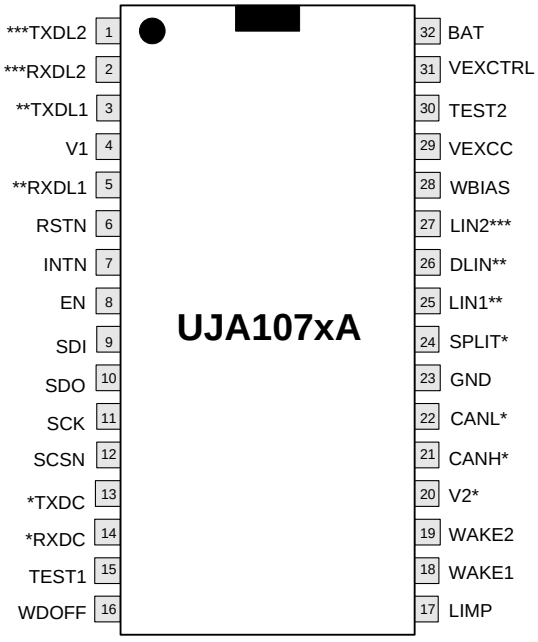


Fig 1. Pinning of UJA107xA variants

2. UJA107xA Details

2.1 UJA107xA Hardware Design

The figure below shows the block diagram of the UJA1078A.

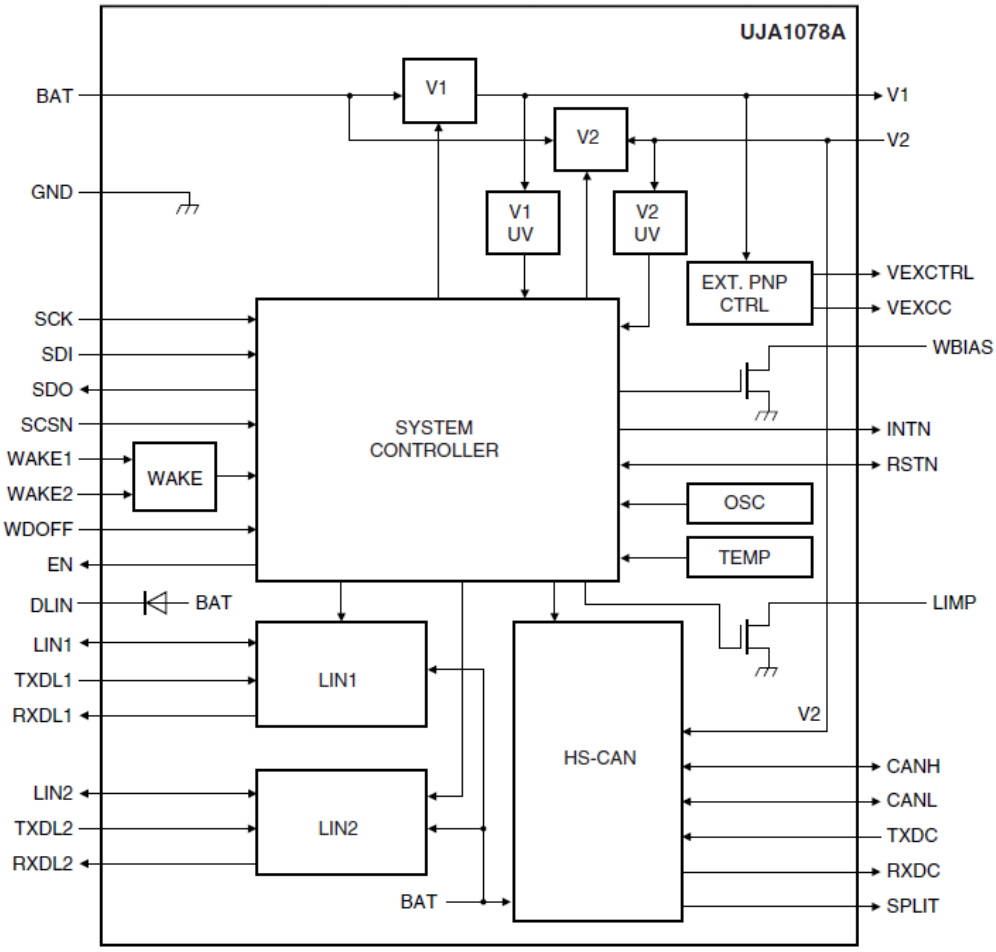


Fig 2. UJA1078A block diagram

The block diagram of the derivatives UJA1075A/76A/79A can simply be derived from this diagram by removing the CAN Transceiver or LIN Transceiver cells, respectively. Furthermore, the related LIN and CAN pins of these derivatives are not usable in the application. Therefore, these pins (RXD, TXD, SPLIT, DLIN and bus pins) can be left open.

The TEST pins should be connected to ground in the application, directly or via a resistor up to 11 kΩ. When an external PNP transistor is used, it makes sense to connect TEST2 via a resistor to GND, because TEST2 is a neighbor pin of VEXCTRL. See also section 2.11.

2.2 Power Supply

2.2.1 Functionality of BAT pin

The BAT pin is the main supply pin of UJA107xA family. The BAT specification is “gap free” starting from 0 V up to 28 V. This can be seen in the figure below. The maximum operating voltage is 40 V. In the supply range between 28 V and 40 V general functionality is still available, but not all parameters might be within the specified limits. In particular, the V1 and V2 regulators, CAN, µC interface, digital control, watchdog and V1 undervoltage monitor work properly up to 40 V.

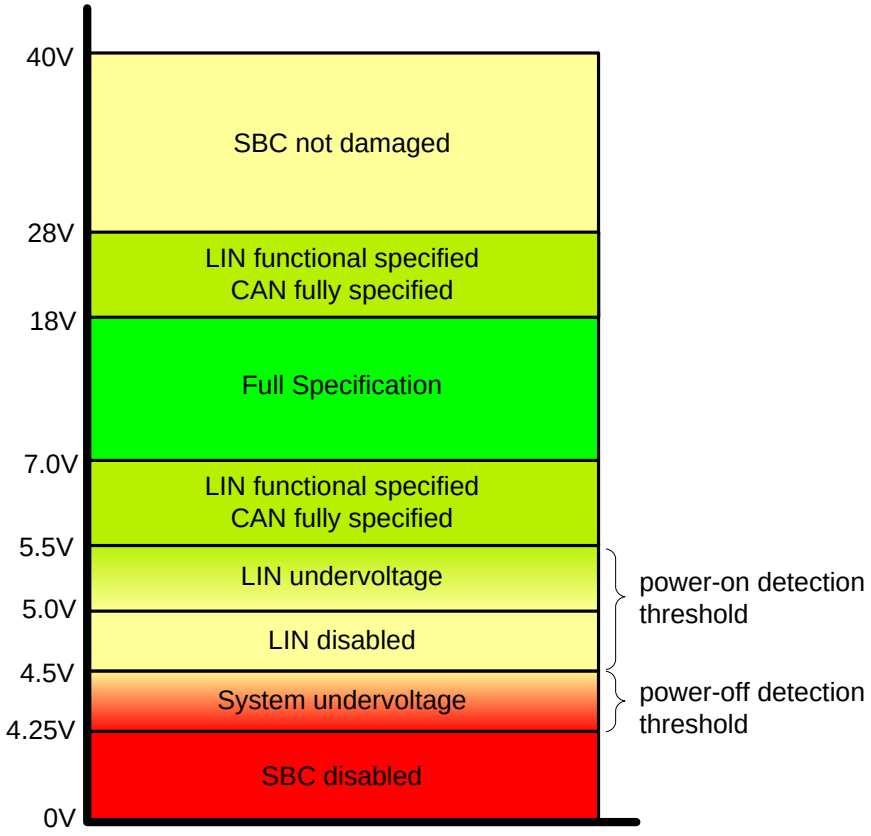


Fig 3. Supply “Gap Free” operation

The main electrical parameters with respect to the BAT pin are the power-on and -off detection thresholds. The UJA107xA is disabled as long as the BAT voltage is below 4.5 V to 5.5 V ($V_{th(det)pon}$). On the other hand it will be disabled when the BAT voltage falls below 4.25 V to 4.5 V ($V_{th(det)poff}$).

At each power-on event of the UJA107xA a dedicated interrupt bit is set to indicate the power-on event to the software. The indication is done by the Power-On Status Interrupt (POSI) which can be found in the Interrupt Status Register (ISR).

Fig 4 shows the recommended minimum BAT circuitry. First of all a reverse polarity protection diode is required because the UJA107xA family doesn't provide any internal reverse polarity protection circuits. Moreover, some buffer capacitors should be applied to the BAT pin to suppress spikes, noise and BAT voltage drops. For reduction of high-frequency noise the small capacitor should be located close to the SBC BAT pin. Depending on the actual requirements regarding voltage transients on the car battery, it may be necessary to add a unidirectional suppressor diode (e.g. PTVS33VP1UP) between BAT pin and GND, or a bidirectional suppressor between battery signal and GND.

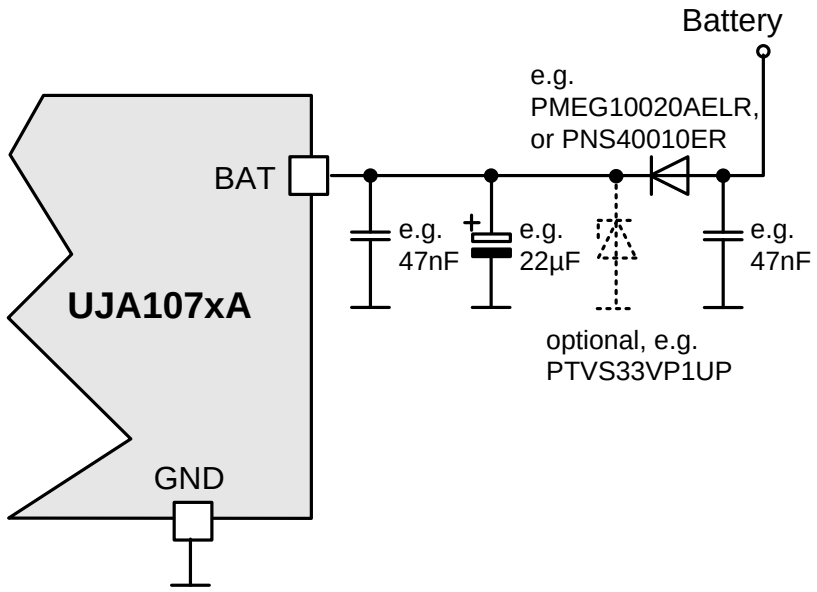


Fig 4. Recommended BAT circuitry

2.2.2 Test Pulses (Cranking Pulse)

The start test pulses defined in ISO 16750-2 [10] and older versions of ISO 7637 [11] simulate battery voltages during engine start (cranking). The figures below illustrate the Test Pulse 4 (Fig 5), Test Pulse 4b (Fig 6) and the Starting Profile of ISO 16750-2 (Fig 7). The waveforms of the test pulses consist of an initial voltage drop to a very low voltage level U1 for less than 100 ms, which is followed by an increase to a slightly

higher level U2 for several seconds. The main difference between these pulses is that the voltage drops on the 4b Pulse and the ISO 16750-2 starting pulse are deeper compared to Pulse 4. If the system must be built in a way that the UJA107xA shall not be disabled during these test pulses and no V1 undervoltage reset is triggered, this can be achieved by choosing the right:

- Reverse polarity protection diode
- Capacitor at BAT pin
- Reset threshold of UJA107xA
- Microcontroller (5 V or 3.3 V)

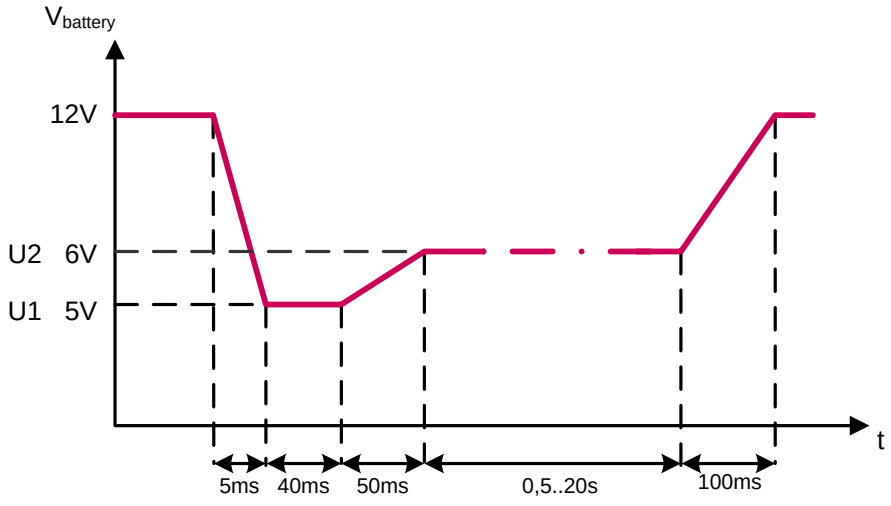


Fig 5. Test pulse 4 in a 12V system

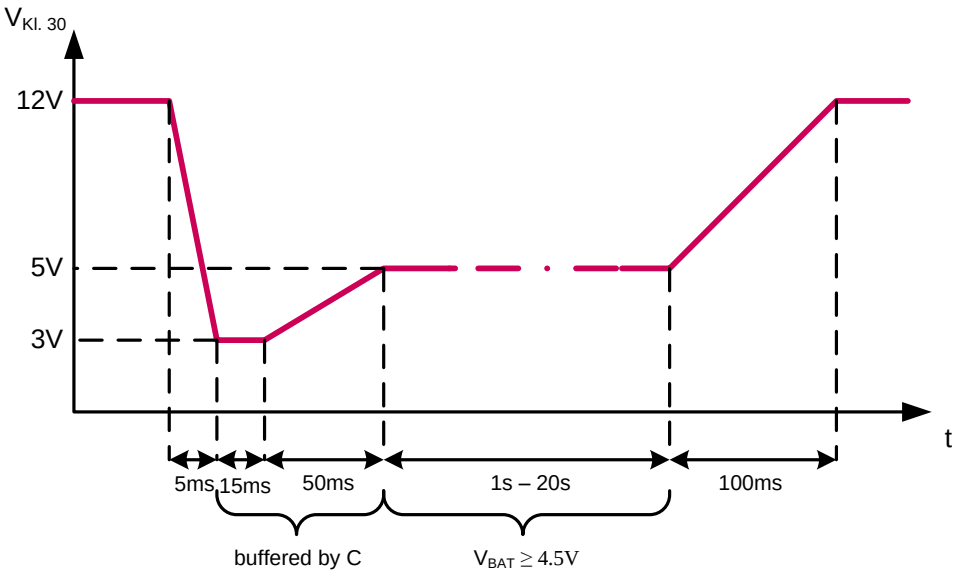


Fig 6. Test pulse 4b in a 12V system

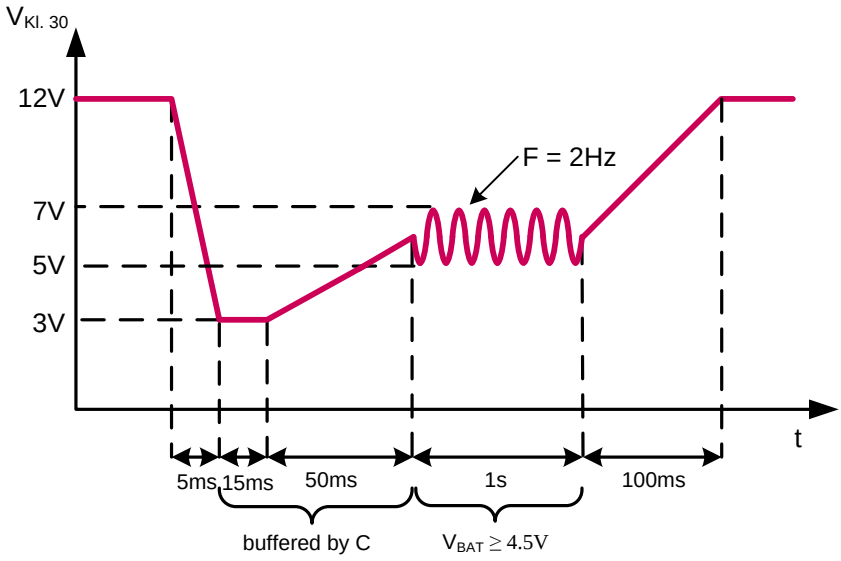


Fig 7. Test pulse (starting profile) ISO 16750-2 in a 12V system

The supply voltage for the SBC at its BAT pin is lower than the battery voltage because of the voltage drop across the polarity protection diode. Moreover, it follows the battery waveform with a delay because of the buffer capacitor at the BAT pin. If the capacitance is big enough, the initial drop of the test pulses can be bridged completely (see Fig 8), so that only the level U2 is the relevant level for the choice of the polarity protection diode and of the undervoltage reset threshold for V1.

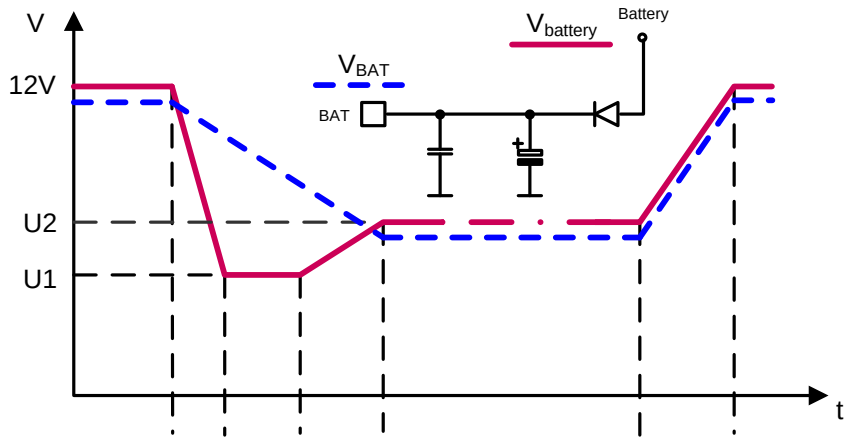


Fig 8. Bridging the initial voltage drop of pulse 4, by using an appropriate buffer capacitor

The table below summarizes, which components are required for each Test Pulse. A more detailed analysis is given in the following subchapters.

Table 3. Test Pulse component consideration

Component	Test Pulse 4	Test Pulse 4b	ISO 16750 starting profile
Reverse polarity protection diode	PN-diode or Schottky diode	Schottky diode	Schottky diode
C at BAT pin	Capacitor voltage may decrease by 6V during 95ms.	Capacitor voltage may decrease by 7V during 70ms.	Capacitor voltage may decrease by 7V during 70ms.
Reset threshold (5V µC)	90% or 70% depend on µC current and diode type	70%	70%
Microcontroller	5V or 3.3V	5V or 3.3V	5V or 3.3V

2.2.2.1 Test Pulse 4

Since the U2 level of Test Pulse 4 is at 6 V, a conventional PN-diode with a maximum voltage drop of 1 V can be used. The resulting supply voltage at the BAT pin is 5 V and this way it is always higher than the power-off threshold ($V_{th(det)poff}$ = 4.25 V - 4.5 V) of the UJA107xA.

On the other hand, in case of a 5V SBC, the V1 regulator drop must be less than 5 V – 4.75 V = 250 mV, when the 90% (ICR_RTHC = 0) reset threshold (V_{uvd} = 4.5 V - 4.75 V) shall be used. Therefore the microcontroller can consume up to

$$\frac{250mV}{R_{(BAT-V1)}} = \frac{250mV}{3\Omega} = 83mA.$$

If more than 80 mA are required during Test Pulse 4, the reset threshold must be changed to 70% (ICR_RTHC = 1), or it must be taken into account to use a 3.3 V microcontroller. Using a 3.3 V microcontroller in cranking applications will only help with respect to more allowed current consumption. As the power-off threshold of the UJA107xA is still at max 4.5 V ($V_{th(det)poff}$), the capacitor at BAT pin is always required.

For selection of the capacitor value, the BAT supply current and the current of any other circuitry that is connected to the BAT pin are relevant. The BAT supply current is mainly determined by the microcontroller supply current (V1) plus the SBC's own consumption and, if enabled, the CAN supply current. During the initial voltage drop of the test pulse, which lasts T = 95 ms, the capacitor voltage may decrease by ΔU = (12 – 6) V = 6 V

For example, if the overall current is I = 50 mA, the minimum capacitor value is calculated to:

$$C = \frac{I \times t}{\Delta U} = \frac{50mA \times 95ms}{6V} = 792\mu F$$

For getting lower capacitance values it is obviously beneficial to reduce the module's supply current during the initial phase of the engine start, e.g. by stopping CAN communication and/or by reducing the microcontroller's clock speed. On the other hand, it can be checked if the module actually has to stay active during the whole engine start or if a power-on reset at the initial voltage drop is allowed.

2.2.2.2 Test Pulse 4b

Since the U2 level of Test Pulse 4b is at 5 V, a conventional PN-diode with a maximum voltage drop of 1 V cannot be used anymore, but a Schottky diode with a maximum voltage drop of 0.5 V can still be used. The resulting supply voltage at the BAT pin is 4.5 V, which is the same like the maximum power-off threshold $V_{th(det)poff}$ of the UJA107xA.

If a 5 V microcontroller is used, the 70% reset threshold has to be selected, provided that the microcontroller specification allows it. Otherwise it must be taken into account to use

a 3.3 V microcontroller. The 90% threshold cannot be used for 5 V microcontrollers because the V1 voltage is below the 90% undervoltage detection level.

For selection of the buffer capacitor the same considerations apply like for the Test Pulse 4 (see above), this time with the parameters T = 70 ms and ΔV = 7 V.

For example, if the overall current is I = 50 mA, the minimum capacitor value is calculated to:

$$C = \frac{I \times t}{\Delta U} = \frac{50mA \times 70ms}{7V} = 500\mu F \ .$$

2.2.2.3 ISO 16750-2 starting profile

The U2 battery level of the ISO 17650-2 test pulse is a sinusoidal signal around 6 V with F = 2 Hz and an amplitude of 1 V. Hence, the min. U2 level is at 5 V and a conventional PN-diode with a maximum voltage drop of 1 V cannot be used. Rather, a Schottky diode with a maximum voltage drop of 0.5 V, as recommended for test Pulse 4b, should be used. The resulting supply voltage at the BAT pin is minimum 4.5 V, which is the same like the maximum power-off threshold $V_{th(det)poff}$ of the UJA107xA.

If a 5 V microcontroller is used, the 70% reset threshold has to be selected, provided that the microcontroller specification allows it. Otherwise it must be taken into account to use a 3.3 V microcontroller. The 90% threshold cannot be used for 5 V microcontrollers because the V1 voltage is below the 90% undervoltage detection level.

For selection of the buffer capacitor the same considerations apply like for the Test Pulse 4b. (see above)

2.2.3 Power on/off timing

Fig 9 shows an example power-on scenario. The plots show the waveform with the signals

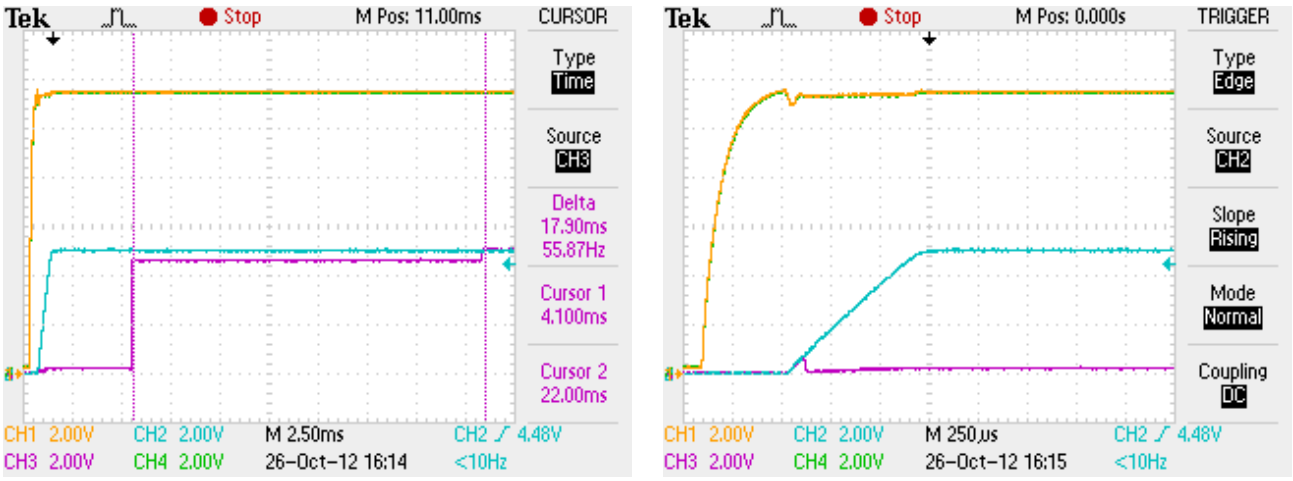
CH1 (yellow) = BAT

CH2 (blue) = V1

CH3 (magenta) = RSTN

CH4 (green) = LIMP

The test set-up includes a 1 kΩ pull-up resistor from RSTN to V1, and a 10 kΩ pull-up from LIMP to BAT.



a. Full waveform

b. Detail of waveform

Fig 9. Power-on timing

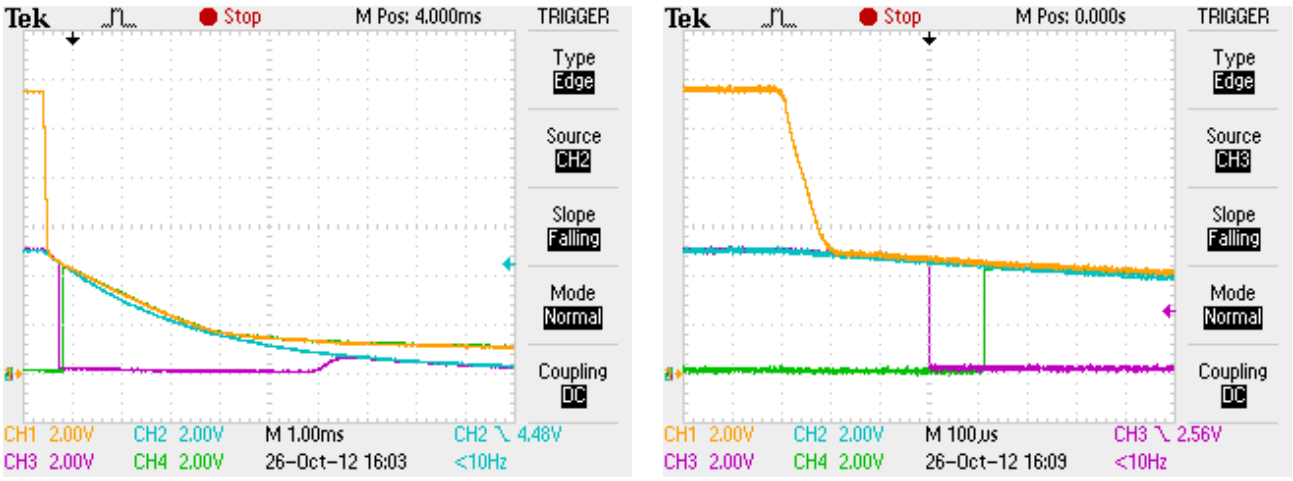
When the BAT level reaches the power-on detection threshold voltage $V_{th(det)pon}$,the SBC starts up and short time later V1 is turned on. The slew rate of the rising V1 voltage is determined by the short-circuit output current $I_{O(sc)}$, by the capacitance of the V1 output capacitor and by the output load. The latter was represented by a 68 Ω resistor in this example.

When the level at V1 has reached the undervoltage recovery voltage V_{uvr}, the timing for the reset pulse length starts. Because of the external pull-up resistor the RSTN signal rises already after the short reset pulse width $t_{w(rst)}$.(left cursor in Fig 9a) The full release of the RSTN signal after the long reset pulse width is also visible, when the RSTN voltage reaches the V1 level (right cursor in Fig 9a).

During power-on the LIMP pin is not driven by the SBC and the pull-up resistor lets LIMP follow BAT.

Fig 10 shows an example power-off scenario. When the supply voltage is turned off, the BAT signal decreases. In this example a very low BAT buffer capacitance was used (only about 50 nF), in order to get the full power-off waveform into a single plot with a focus on the details of V1, RSTN and LIMP. When BAT gets close to the nominal output voltage of V1 (in this case 5 V), the V1 output voltage also starts to decrease. As soon as V1 has crossed the undervoltage detection voltage V_{uvd}, RSTN goes low after the filter time t_{fltr}. When BAT has reached the power-off detection threshold voltage $V_{th(det)poff}$, the SBC turns off. RSTN is pulled low until V1 is below 1 V.

In this example the LIMP pin had been activated before turning off the supply. The SBC de-activates the LIMP signal as soon as BAT has reached the power-off detection threshold voltage $V_{th(det)poff}$.



a. Full waveform

b. Detail of waveform

Fig 10. Power-off timing

2.3 Microcontroller Interface

The output voltage levels as well as the input voltage thresholds of all microcontroller related pins are fully adapted to V1. Thus, if a 3.3 V derivative is used, the complete interfacing between the SBC and the microcontroller operates on a 3.3 V basis.

2.3.1 Reset Interface

2.3.1.1 Functionality of RSTN pin

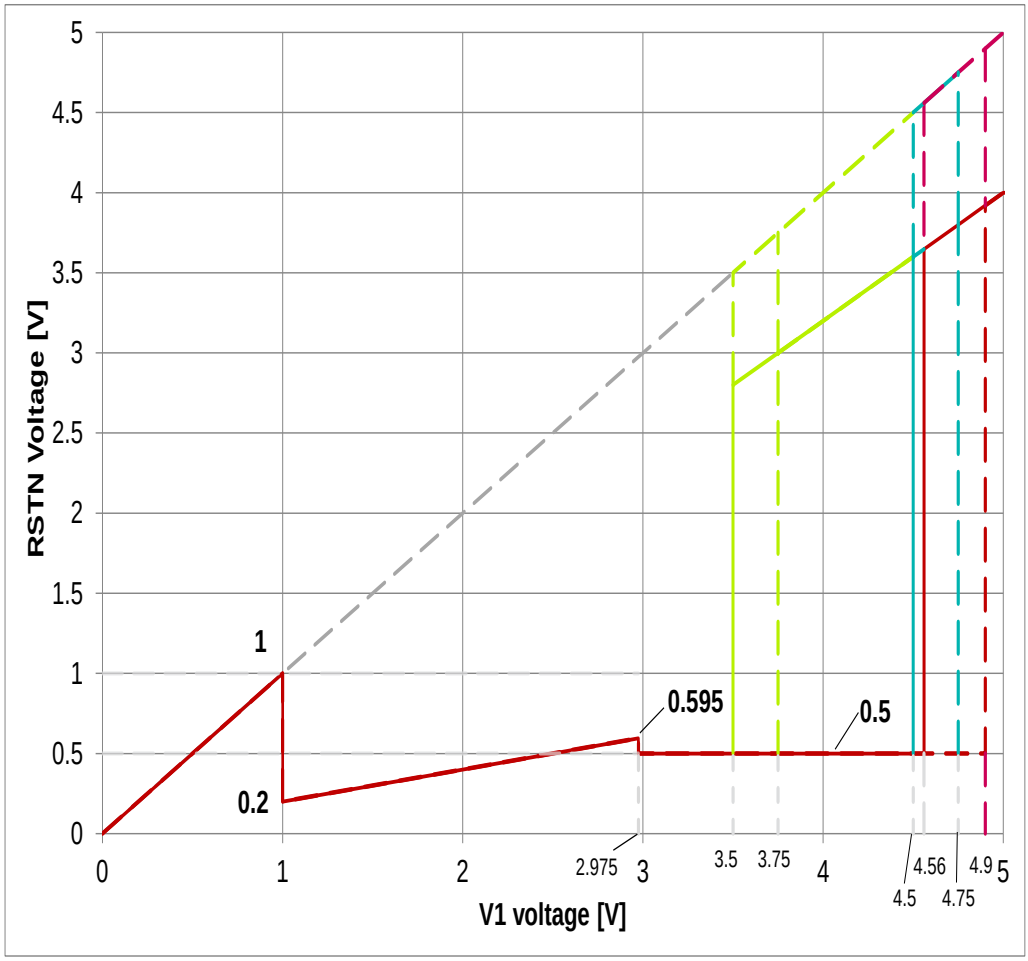
The SBC UJA107xA offers the bidirectional RSTN pin for triggering a system reset. On the one hand the reset of the SBC is triggered if the RSTN pin is forced active low externally e.g. by the microcontroller. On the other hand the SBC generates a reset pulse on the RSTN pin on power-on or any system condition that requires a system reset (see data sheet [1]).

In general there are three different kinds of possible resets that demand different software handling:

1. Reset at power-on: The complete system is not configured. Therefore, the microcontroller and UJA107xA must be initialized.
2. Reset at Wake-up out of Sleep, at overtemperature shutdown or at V1 undervoltage: As the microcontroller was not supplied in these cases, its configuration is lost and hence it must be completely reinitialized.
3. Reset at WD failures, Software Resets, external resets: As these resets are caused by SW failures the application must be checked and reinitialized.

In Sleep or Overtemp mode the RSTN pin is always LOW and becomes HIGH again after either the chip is cooling down or a wake-up event is detected. Moreover, the RSTN

pin is only released when V1 has left its undervoltage state and the reset length timer (see chapter 2.3.1.2) has elapsed. In that process it must be considered that the reset length timer is not started before the V1 voltage is above its 90% undervoltage threshold (4.5 V – 4.75 V). This behavior is independent of the RTHC bit configuration in the Interrupt Control Register



Red: V1 rising; Turquois: V1 falling, 90% threshold; Green: V1 falling, 70% threshold

Fig 11. Gap free RSTN pin specification

The RSTN pin voltage is defined by the V1 voltage level and is completely independent of the battery supply. Its gap-free characteristics for 5 V V1 are illustrated in Fig 11 (with the condition of a pull-up resistor to V1 of min. 900 Ω):

- If V1 < 1V, the voltage at RSTN is at most equal to V1 voltage (area 1 in figure above).
- If 1 V < V1 < 2.975 V, the voltage at RSTN is at most 0.2 x V1 (max. 0.6 V) (area 2 in figure above).

- If V1 > 2.975 V the RSTN pin is forced to less than 0.5 V by the internal reset driver (area 3 in figure above).
- When V1 rises above 4.56 – 4.9 V, RSTN gets high to at least 0.8 x V1 after the reset pulse width time has elapsed
- When the 90% threshold has been selected and V1 falls below 4.5 – 4.75 V, RSTN is pulled to max 0.5 V
- When the 70% threshold has been selected and V1 falls below 3.5 – 3.75 V, RSTN is pulled to max 0.5 V

This ensures that in case of low V1 voltage the RSTN pin voltage is always lower than 1 V. Hence, it prevents an unsupervised operation of the microcontroller as it remains in reset state.

2.3.1.2 Configuration of reset length

The UJA107xA provides a selectable startup reset length for the microcontroller. Startup means that the microcontroller was unsupplied. This is the case at any Power-On, V1 undervoltage, Overtemp and Wake-up out of Sleep Mode. These reset events can trigger a short or a long reset pulse. All other UJA107xA resets like WD failure, external and software resets always trigger a short reset pulse. The selection of the reset pulse length is done by using an external pull-up resistor on the RSTN pin (see Fig 12). A short reset pulse length ($t_{w(rst)}$ short = 3.6 ms – 5 ms) is selected by connecting a 1 kΩ ±10 % resistor between the pins RSTN and V1. If no resistor is connected, the reset pulse will be long ($t_{w(rst)}$ long = 20 ms – 25 ms).

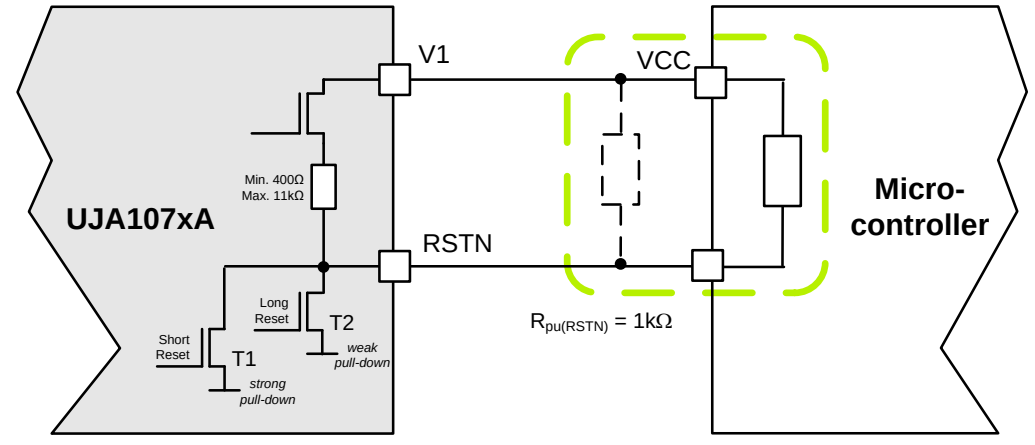


Fig 12. Reset length configuration

The RSTN pin has two internal pull-down transistors. In case the external pull-up is stronger than the internal pull-down transistor T2 (weak pull-down), there is always a short reset pulse length. Therefore, if the microcontroller has an internal pull-up resistor

at the reset pin, its value must be at least 25kΩ to allow selecting the long reset. This ensures that the RSTN pin level is below 0.5V for the whole reset pulse length ($t_{w(rst)}$ long = 20ms – 25ms).

For example if the microcontroller has an internal pull-up of e.g. 30kΩ, the short reset pulse is selected by connecting an additional 1kΩ pull-up resistor. If no external resistor is connected, the reset pulse will be long.

Due to the transition from strong to weak pull down a small RSTN voltage increase will be visible when the reset is driven low. This is caused by deactivating transistor T1 (strong pull-down) and having only the weak transistor T2 active after $t_{w(rst)}$ short reset pulse length. Nevertheless, the voltage at the RSTN pin is always lower than 0.5 V for the whole reset pulse length. Fig 13 illustrates that behavior for e.g. a power-on reset under normal operating condition (V_{BAT} > 5.5 V) without an additional 1 kΩ pull-up resistor (long reset pulse).

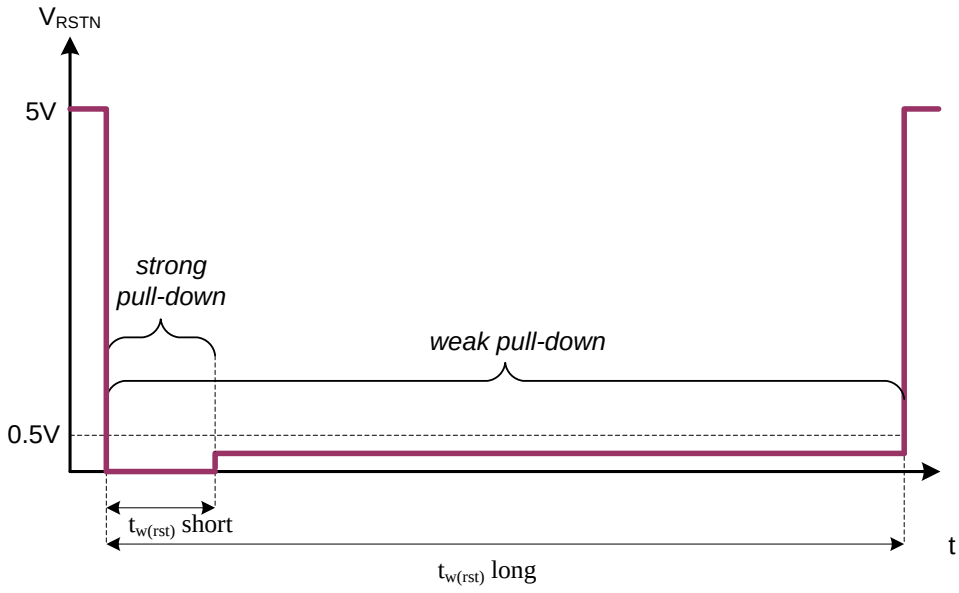


Fig 13. RSTN pin behavior at long reset pulse

When the short reset is selected the RSTN pin behavior is illustrated in Fig 14. This configuration requires a 1 kΩ pull-up resistor between the pins RSTN and V1. Due to the pull-up resistor the RSTN voltage will not become 5 V before the weak pull down transistor (T2) in the SBC is disabled. The voltage divider of the pull-up resistor and the weak pull down resistor determine the voltage level at the RSTN pin during that time.

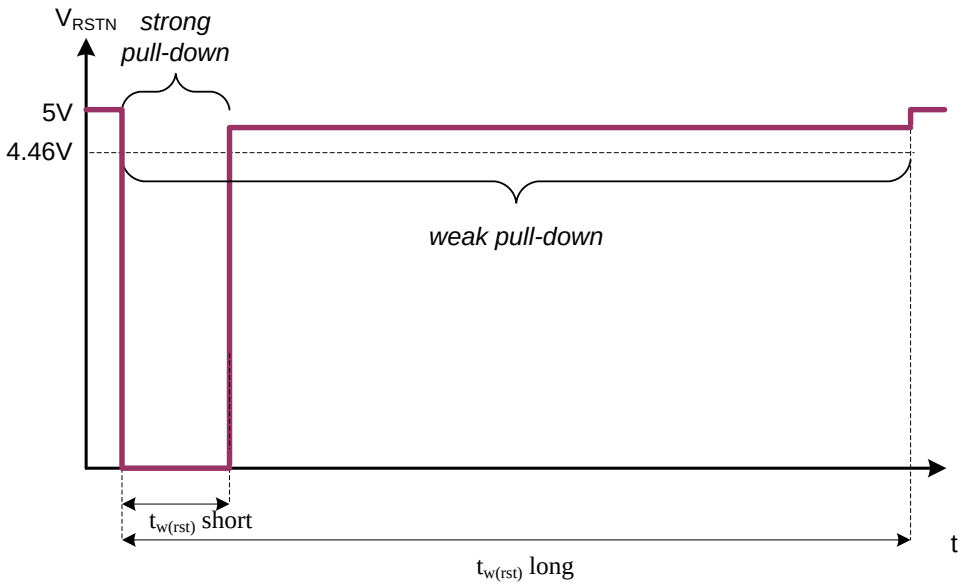


Fig 14. RSTN pin behavior at short reset pulse

2.3.2 Interrupt Interface

The interrupt interface is used to inform the software about any event related to the SBC, e.g. voltage regulator failures or wake-up events. Moreover, it is recommended to connect the INTN pin to IRQ/XIRQ pin of the microcontroller or at least to a pin with external interrupt capabilities.

2.3.2.1 Functionality of the INTN pin

The INTN output pin is an active low, open drain interrupt signal line that is used to inform the microcontroller about SBC events. The interrupt pin can be wired-or connected with interrupt lines of other sources. If no other component provides a pull-up resistor at the interrupt line, a pull-up resistor (e.g. 10 kΩ) to V1 has to be applied.

The INTN pin is forced low when at least one interrupt is pending. It is released only if all bits in the Interrupt Status Register (ISR) are cleared by the user. If all interrupts are cleared, the INTN pin is set floating and thus, the interrupt signal becomes immediately high again due to the external application pull-up resistor.

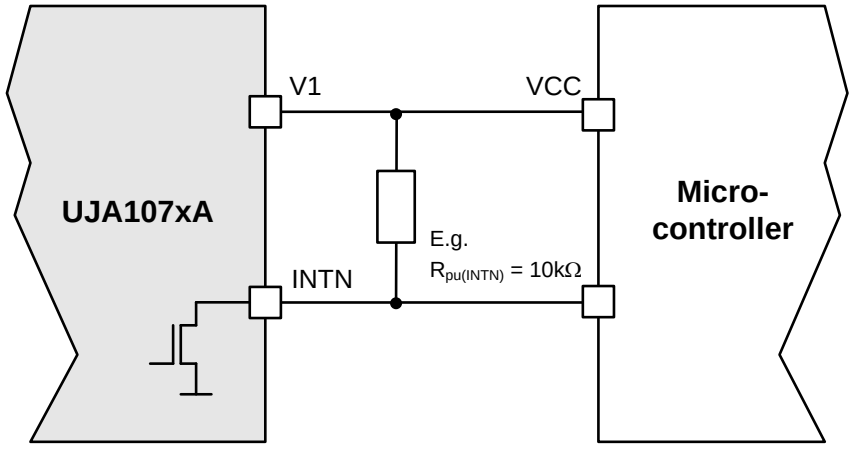


Fig 15. INTN Interface

2.3.2.2 Interrupt enable/clear mechanism

All interrupt sources of the SBC can be separately enabled and disabled in the Interrupt Control Register (ICR) except of the Power-On Status Interrupt (POSI) bit, which is always enabled. The status of the interrupt sources can be read in the Interrupt Status Register (ISR). If a bit is “1”, the according interrupt is set. Reading interrupt status bits does not change the content and thus, interrupt status bits can be read multiple times. Pending interrupts have to be cleared actively by writing “1” to the dedicated bit in the Interrupt Status Register. Writing “0” to the Interrupt Status Register has no effect.

2.3.3 SPI Interface

The SPI interface is the main communication channel between SBC and microcontroller. By use of the SPI interface the microcontroller configures the SBC, reads back status information of the SBC and triggers the watchdog.

2.3.3.1 Functionality of the SPI pins

The SBC is controlled via the 4-wire SPI interface as shown in Fig 16. The interface is a 16bit SPI that consists of 4 digital pins: Serial Data In (SDI), Serial Data Out (SDO), SPI clock in (SCK) and SPI Chip Select (SCSN). These connections can optionally be applied with e.g. 1 kΩ series-resistors that may help to reduce EMC emission, especially if there is a long distance between the host controller and the SBC.

To ensure that the SCSN pin and the SCK pin are always on a defined level the SBC contains an internal pull-up resistor to V1 at SCSN and an internal pull-down resistor to GND at SCK. Pin SDO is driven by an internal push-pull output that is only active if the SCSN pin is LOW. Otherwise the SDO pin is floating. Therefore, it is possible to connect another SPI device in parallel if it has an own chip select pin.

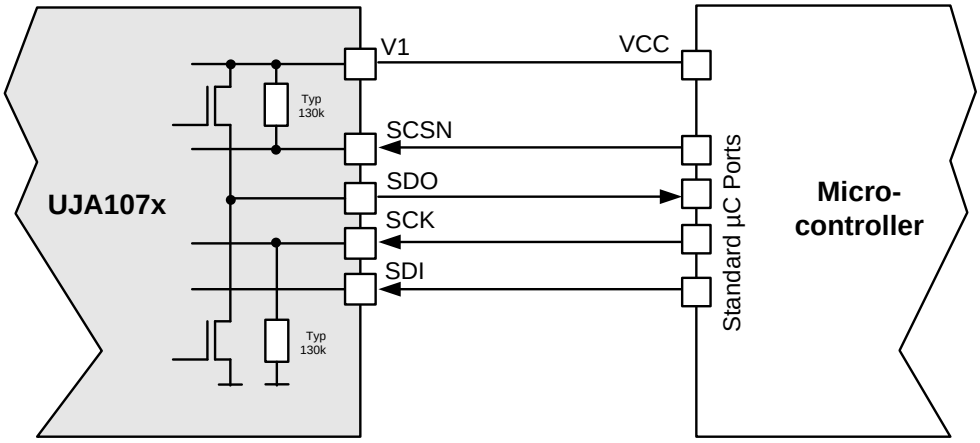


Fig 16. SPI interface

Fig 17 shows an example configuration when an additional SPI device is connected in parallel to the same microcontroller SPI interface. Except for the SCSN all pins can be shared between the SBC and the second slave, provided that the SDO of the second slave is also floating when its SCSN is HIGH. This configuration can be extended by other SPI devices that also have SDO pins with the same characteristic.

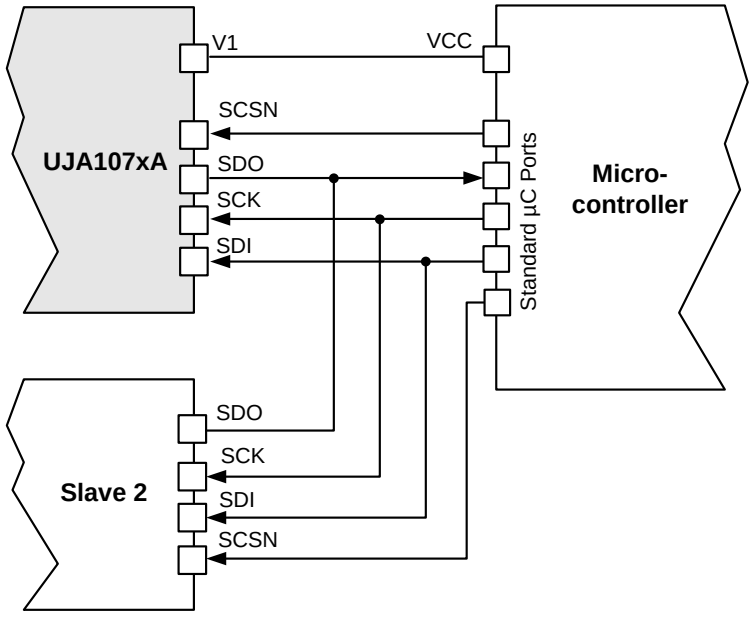


Fig 17. SPI configuration if two slaves are connected to the same MCU SPI Interface

2.3.3.2 Configuration of SPI Interface

The UJA107xA family provides a 16-bit SPI Interface. The SBC tolerates also SPI messages with more than 16 bits but only the first 16 bits are considered. SPI messages with less than 16 bits are completely ignored.

Fig 18 shows how the SPI interface has to be configured. The SBC shifts data with the rising edge and samples with the falling edge of the pin SCK. The initial setting of SCK when SCSN goes down is LOW. Furthermore, the SBC expects that the most significant bit is sent first. The SPI is a bidirectional data transfer. As one bit is written into SDI, another bit is shifted out of SDO (see Fig 18).

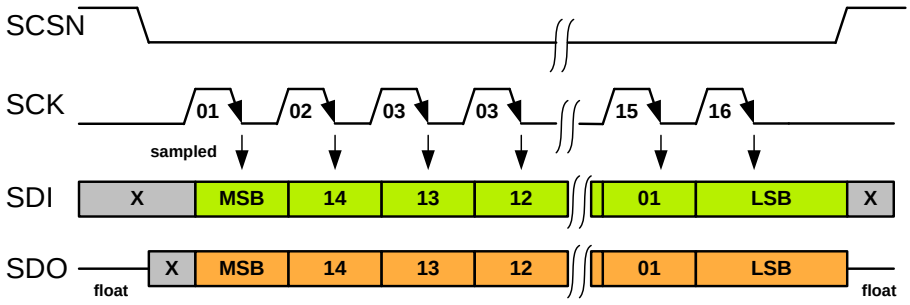


Fig 18. Configuration of SPI interface

2.3.3.3 SPI Register Architecture

The SPI allows for full duplex data transfer. Therefore status information is returned when new control data is shifted in. Fig 19 shows the register structure of the UJA107xA SBC. The upper three bits of the 16-bit SPI message determine which register is addressed. The registers of the SBC are listed in Table 4. Bit 12, the Read Only bit (RO), determines whether something is actually written into the addressed register or not. Hence, this bit allows a read-only access option where registers are read back by the application without changing the register content. If this bit is set to “1” the SPI transfer is a read-only access and all data bits (bit 0 to 11) written into SDI are ignored. If the RO bit is “0”, the data bits 0 to 11 are written into the addressed register. The written data become valid as soon as the SCSN pin returns on HIGH level.

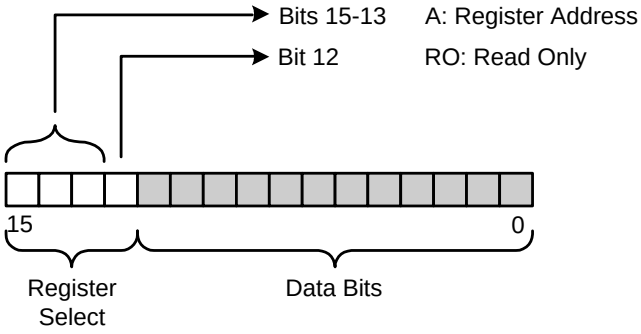


Fig 19. Structure of SPI register

Table 4. Register Map

Address Bits 15 to 13	Data bits 11 to 0 (Read/Write depending on RO)
000	Watchdog and Status register
001	Mode Control register
010	Interrupt Control register
011	Interrupt Status register

2.3.4 Routing of SPI signals on the PCB

The routing of the SPI interconnection lines on the PCB shall ensure that SPI data transfers are without errors. This can be achieved by:

- Using short interconnection lines
- Adding shielding GND lines
- Minimizing the loops of high-frequency signals or signals with high current amplitudes (e.g. fast clock signals, supply lines for inductive loads or any unfiltered outside signal) and maximizing their distance to the SPI lines

The SBC ignores SPI commands with less than 16 bits (i.e. less than 16 clock cycles on SCK) between falling and rising edge of the chip select signal SCSN. Section 3.3.4.1 explains how SPI communication errors can be detected by software. Section 3.3.4.2 explains why the use of shadow register can reduce the impact of SPI communication errors. Nevertheless first priority should be to exclude SPI communication errors by appropriate PCB design.

2.4 Voltage Regulators

Within the UJA107xA two independent low-dropout (LDO) voltage regulators supplied by battery are integrated.

2.4.1 V1 voltage regulator

The main voltage regulator V1 offers the supply for the application microcontroller, its periphery and additional transceivers. Therefore, the V1 regulator can deliver up to 250 mA. The UJA107xA family provides devices with 5 V and 3.3 V V1 regulator. The figure below shows the typical V1 output circuitry.

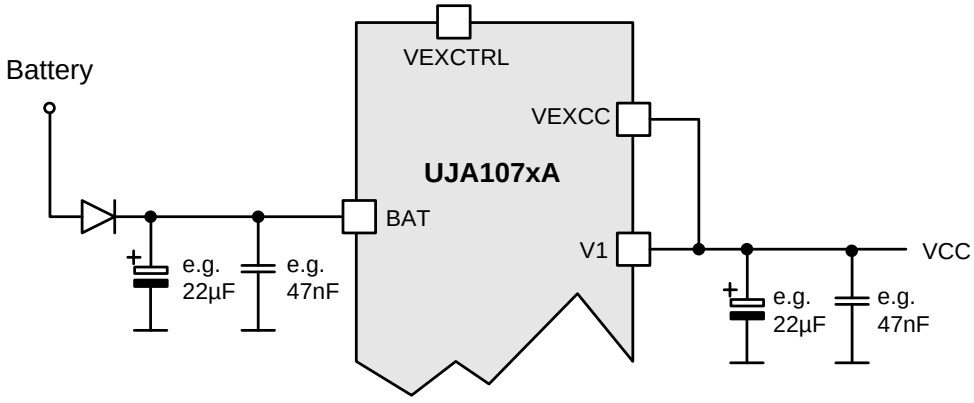


Fig 20. Typical V1 application

In case no external PNP transistor is connected to the SBC (see chapter 2.4.2), the pin VEXCTRL can be left open and pin VEXCC must be connected to V1. When the PCB is prepared for usage of an external PNP as population option, the shunt resistor between V1 and VEXCC needs to stay populated also when no PNP is populated.

2.4.1.1 V1 output capacitor

The value of the V1 output capacitor depends on the requirements for line and load regulation. The V1 regulator is stable with a large range of capacitor values and a large range of ESR values.

As a rule of thumb, Table 5 lists the recommended minimum capacitor value to be applied to the V1 output, and the max ESR, when a tantalum electrolytic capacitor is used. Both values depend on the expected maximum load current. For the 3.3 V version of the V1 regulator a larger capacitance is needed than for the 5 V version.

Higher ESR values mean e.g. higher undershoots upon fast load current increase. On the other hand, when the ESR gets much lower than 100 mΩ, the output voltage will show some small initial ringing during regulation of line or load changes. In case this

needs to be avoided, a higher ESR can be emulated by adding an external 100 mΩ resistor between V1 pin and capacitor(s).

When using a ceramic capacitor with small physical dimensions, note that its actual capacitance may be much smaller than the nominal value. This is not only because of accuracy, temperature and aging, but also because the capacitor is operated with a DC offset voltage of 3.3 V or 5 V. This DC voltage may reduce the available capacitance significantly. For details please refer to the datasheet of the used capacitor. Usually a ceramic capacitor with a much higher value needs to be used in order to keep sufficient capacitance when operated with the nominal output voltage of the LDO.

Table 5. Recommended minimum capacitor value at V1 and max ESR

V1 load current	5 V version	3.3 V version	max ESR
≤ 125 mA	22 µF	68 µF	2 Ω
> 125 mA	68 µF	100 µF	1 Ω

When a capacitor is used with lower capacitance than recommended in Table 5, it needs to be checked if the regulation performance is still good enough. For load regulation the worst case can be found usually at high temperature, because the regulator then gets slower. For line regulation the worst case can be found usually with fast rising edges starting between 4.25 V and 6 V, and at low temperature, because the re-activation after overshoot limitation (see section 2.4.1.4) then takes more time. In general it is recommended to avoid a combination of capacitance lower than 22 µF with ESR < 100 mΩ.

2.4.1.2 V1 low-power modes

The V1 voltage regulator is active in Standby and Normal Mode. In Sleep Mode it is switched off in order to achieve lowest quiescent current. Beside this V1 is also disabled in Off Mode and Overtemp Mode to protect the device and to take care that the system is not running in an unstable condition.

The regulator consumes only very little supply current. That current is included in the value I_{BAT}, which the datasheet provides for the condition I_{V1} = 0 mA. When the output current is not zero, I_{BAT} increases by that amount of output current plus a small additional current of about 1.2 µA per 100 µA output load (i.e. 1.2%), which the regulator consumes itself.

2.4.1.3 V1 undervoltage monitor

If V1 is active, it is monitored by an internal undervoltage detector. In case V1 voltage drops below the undervoltage threshold, the SBC pulls down the RSTN pin. The UJA107xA family offers two configurable reset thresholds, one at 90% and another one at 70% of the nominal V1 voltage (V_{uvd}). The reset threshold is controlled by the Reset Threshold Control (RTHC) bit in the Interrupt Control Register (ICR). The threshold should be selected according to the specified operating range or undervoltage detection threshold of the microcontroller. The UJA107xA undervoltage detection threshold must be above the related microcontroller thresholds.

If the 70% undervoltage reset threshold is selected, a pre-warning interrupt (V1UI – V1 Undervoltage Interrupt in the Interrupt Status Register) can be activated that performs an interrupt if V1 drops below the 90% undervoltage threshold. Beside this also the V1 Status bit V1S shows whether V1 is above the 90% threshold or not. In case the V1 undervoltage threshold is configured to 70% of the nominal V1 voltage, the RSTN pin is not released before the V1 voltage has reached the 90% nominal voltage (V_{uvr}) and the reset length timer has elapsed (for more information see 2.3.1.1).

2.4.1.4 V1 overvoltage de-activation

The V1 regulator has a built-in overvoltage protection feature, which quickly turns off V1, when the output voltage exceeds a threshold level between 5.25 V and 5.6 V. The de-activation time is in the order of 1 µs. The LDO is re-activated automatically about 80 µs after the output voltage has returned below the threshold. During that time the load current is supplied only by the output capacitor and therefore the capacitor gets discharged and its voltage decreases. Then it still takes a few microseconds for ramping up the LDO output current so that the capacitor voltage stops falling and rises again back to the target level.

Accordingly, the V1 output capacitance needs to be high enough for keeping the output voltage within the desired range upon such overvoltage event.

2.4.2 V1 voltage regulator plus external PNP transistor

The V1 regulator can deliver up to 250 mA by itself. To prevent overheating of the device in case of high ambient temperature or high average currents, it is possible to connect an external PNP transistor as shown in Fig 21. In this configuration the power dissipation of the V1 regulator is distributed between the SBC and the PNP transistor. The external PNP transistor will be switched on when the load current is above the selected power distribution threshold ($I_{th(act)PNP}$) and the battery voltage is higher than 5.9 V - 7.5 V ($V_{uvd(ctrl)lext}$). If the battery voltage decreases below that threshold, the PNP transistor is not disabled immediately. The current delivered by the PNP transistor is not reduced until the load current is lowered.

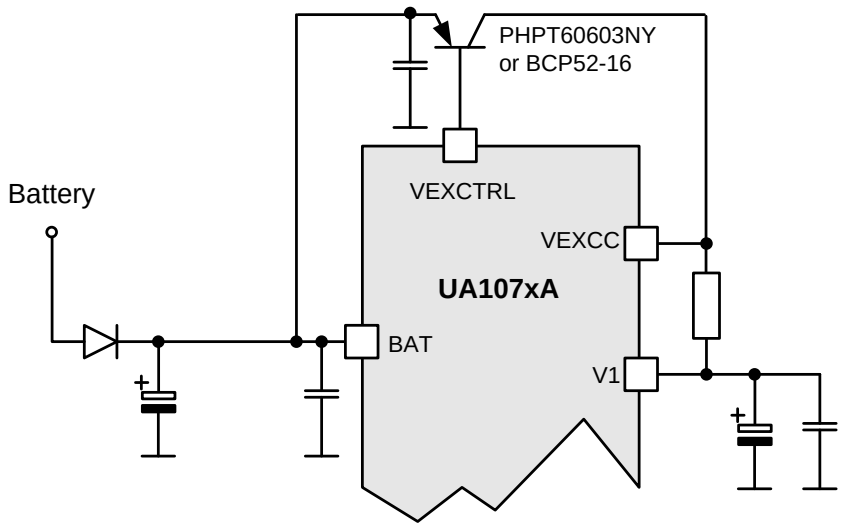


Fig 21. External PNP circuitry

The shunt resistor between V1 and VEXCC is required for measuring the current of the transistor. This is needed for its short-circuit protection. In case the voltage between VEXCC and V1 becomes higher than $V_{th(act)lim}$, the PNP current is not increased anymore. For calculating the resistor value, the resistor voltage drop caused by the PNP current ($V_{th(act)Ilim}$), the ambient temperature and the thermal performance of the transistor on the PCB have to be considered (e.g. for a BCP52-16 PNP a resistor of at least 5.6 Ω is recommended that limits the current delivered by the PNP transistor to $V_{th(act)Ilim\_max}$ / 5.6 Ω =59 mA).

For test purposes the transistor type BCP52-16 (NXP Semiconductors) was used. However, the selection of the PNP is not critical. In general any PNP with a current amplification factor β between 60-500 can be used.

Close to the PNP a small capacitor (10 – 100 nF) from emitter to GND should be connected. This prevents turning on the PNP by injected RF power during low-power mode of the SBC. A pull-up resistor between Emitter and Base is not necessary, because the VEXCTRL pin already pulls high when the transistor shall be turned off. In case such external resistor is used anyway, part of the current provided by the VEXCTRL pin will flow to that resistor, leaving a bit less current for the base and thus requiring a slightly higher minimum amplification factor of the transistor.

One advantage of this scalable voltage regulator concept is that there are no specific PCB layout restrictions for using the external PNP. The distance between the UJA107xA and the external PNP has no negative impact on the stability of the regulator loop because the control loop for the PNP is running with a relatively low clock frequency of 8 kHz. Hence, there is no influence of signal resonances. Therefore, it is recommended to increase the distance between UJA107xA and PNP to its maximum for a better thermal distribution on the PCB.

The influence of the V1 output capacitor on the regulation performance, as discussed in section 2.4.1, is still the same, when a PNP is used. The response to fast load steps is mainly determined by the V1 regulator performance. Afterwards the SBC adjusts the PNP current to restore the desired balance between internal V1 output current and PNP output current. When the load current is reduced quickly, it may become lower than the PNP current for a short moment, until the regulator has reduced the PNP current sufficiently. During that time an internal sink current is activated for compensation of the excess PNP current. In other words, while the PNP is active, the internal regulator is a two-quadrant regulator that can drive the output current in both directions.

For adjusting the power distribution between the SBC and the PNP, the UJA107xA family provides the Power Distribution Control Bit (PDC) which can be found in the Mode Control Register (MCR). This bit determines how much current is driven by the internal V1 regulator when the PNP becomes active. The PNP activation thresholds given in the data sheet are temperature dependent. A configuration of PDC=1 means for example that for rising load currents, the SBC delivers typically 50 mA at Tvj=150°C. For lower temperatures the current delivered by the SBC is higher. This temperature dependent behavior prevents an overheating of the SBC, as with increasing junction temperature the PNP activation current threshold is lowered.

Using the SBC together with an external PNP transistor allows distributing the power dissipation of the V1 regulator. This prevents overheating of the device in case of high average currents and high ambient temperature. If we assume a system with an average V1 current of 200 mA and a battery voltage of 16 V the power dissipation of the V1 regulator can be calculated as:

$$P_{V1\_regulator} = (V_{BAT} - V_{V1}) * I_{V1} = 11V * 200mA = 2.2W$$

When an external PNP is added this power can be shared between the SBC and the transistor so that the heat is distributed over a wider area of the PCB:

1. If an external PNP is used and PDC=0, power dissipation distributes as follows:

$$P_{V1\_regulator} = (V_{BAT} - V_{V1}) * I_{V1} = 11V * 85mA = 0.935W$$

$$P_{PNP} = (V_{BAT} - V_{V1}) * I_{PNP} = 11V * 115mA = 1.265W$$

2. If an external PNP is used and PDC=1, power dissipation distributes as follows:

$$P_{V1\_regulator} = (V_{BAT} - V_{V1}) * I_{V1} = 11V * 50mA = 0.55W$$

$$P_{PNP} = (V_{BAT} - V_{V1}) * I_{PNP} = 11V * 150mA = 1.65W$$

The example shows that the V1 power dissipation can drastically be reduced by using an external PNP transistor. Nevertheless, the thermal performance of the transistor must be considered. For condition 1 e.g. a single transistor PHPT60030PY or two transistors BSP31 with a shunt resistor R=2.7 Ω (I_{max}=122 mA) may be used. For condition 2 e.g. one transistor PHPT60030PY with a shunt resistor R=2.2 Ω (I_{max}=150 mA) can be used to ensure the needed collector current.

The table below gives an overview and additional information for other system configurations. The V1 power dissipation of other use cases can be calculated by using the power dissipation calculator attached to the application hints.

Table 6. Examples for distribution of power dissipation

Load current	V1 current	PNP current	V1 power dissipation	PNP power dissipation	Transistor type*	Shunt resistor
200mA @16V	85mA	115mA	0.935W	1.265W	e.g. 1 x PHPT60030PY or 2 x BSP31*	R=2.7Ω
	50mA	150mA	0.55W	1.65W	e.g. PHPT60030PY*	R=2.2Ω
100mA @16V	85mA	15mA	0.935W	0.165W	e.g. BCP52-16*	7.5Ω≤R≤22Ω
	50mA	50mA	0.55W	0.55W	e.g. BCP52-16, BSP31*	R=5.6Ω
150mA @16V	85mA	65mA	0.935W	0.715W	e.g. BSP31*	R=4.7Ω
	50mA	100mA	0.55W	1.1W	e.g. 1 x PHPT60030PY or 2 x BCP52*	R=3.3Ω

* The proposed transistors have characteristics that likely fit to the respective use case example. However, their actual suitability is strongly depending on the thermal conditions (ambient temperature, thermal resistance of the PCB etc.) and still needs to be checked for each board design individually.

In case of high ambient temperatures (105°C or 125°C) and high load current requirements (I_{load}>150 mA) it may be required to use several external PNPs in parallel to reduce the dissipation of each external PNP transistor and/or to achieve even better heat distribution across the PCB. Fig 22 shows how to connect two parallel PNP transistors to the external transistor interface of the UJA107xA. In order to control the distribution of the current between the two PNP transistors, the voltage drop across the emitter resistors R1 and R2 at full current needs to be in the range 0.5 V…1 V. This prevents a thermal runaway, which would otherwise let one transistor take over the full current.

For example, a system with an average total V1 current of 150mA and two BCP52-16 transistors may look like this:

- SBC V1 current 50 mA (PDC = 1)
- PNP transistor 1 current 50 mA
- PNP transistor 2 current 50 mA.

Therefore, both emitter resistors should be R1 = R2 = 10…20 Ω. Furthermore, a shunt resistor R3 = 3.3 Ω for the limitation of the sum current of the external PNPs is recommended.

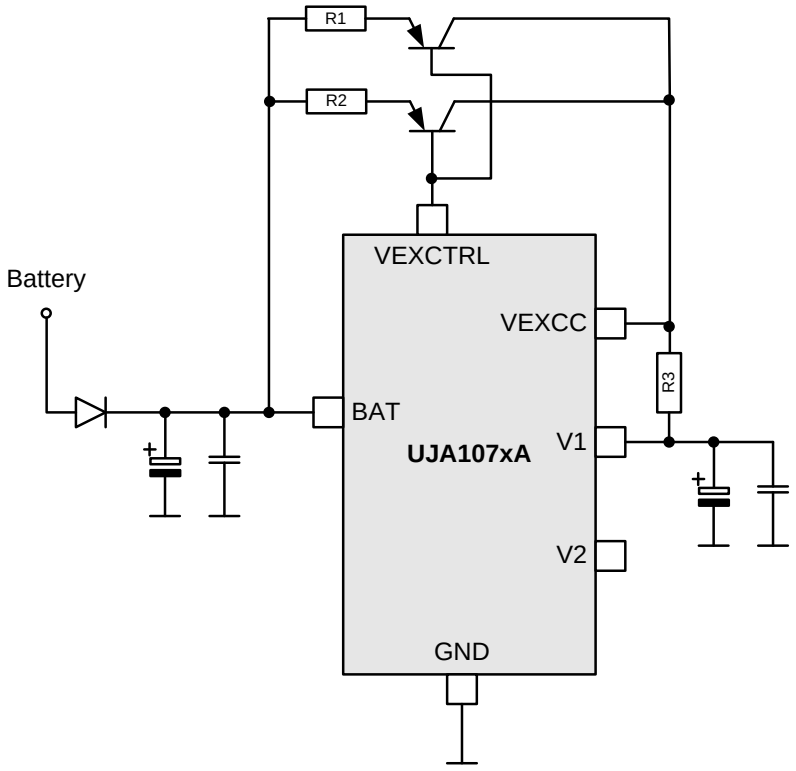


Fig 22. External circuitry for two PNP transistors in parallel

Please note, that VEXCTRL pin is implemented as current source and therefore cannot directly be used to control MOSFETs.

2.4.3 V2 voltage regulator

2.4.3.1 Functionality of the V2 regulator

The second voltage regulator V2 is a 5 V regulator dedicated for supplying the integrated CAN Transceiver. This ensures that the microcontroller supply V1 is separated from the CAN interface. The V2 output is primarily used to connect a capacitor, which buffers the CAN Transceiver supply, as shown in Fig 23. It is recommended to add a minimum capacitance of 10 µF at V2 to allow undisturbed CAN communication. For stability of the V2 regulator the ESR of the capacitor should be in the range of 500 µΩ to 3 Ω.

When the V2 output current is high while the supply voltage at the BAT pin rises fast with a start value between 4.25 V (i.e. above the power-off voltage) and 5.5 V (i.e. while the LDO is in drop-out mode), a voltage overshoot at the V2 pin might occur that exceeds its

max ratings. Therefore a limitation of the BAT pin voltage slew-rate to maximum 0.5 V / µs is necessary for rising edges starting between 4.25 V and 5.5 V.

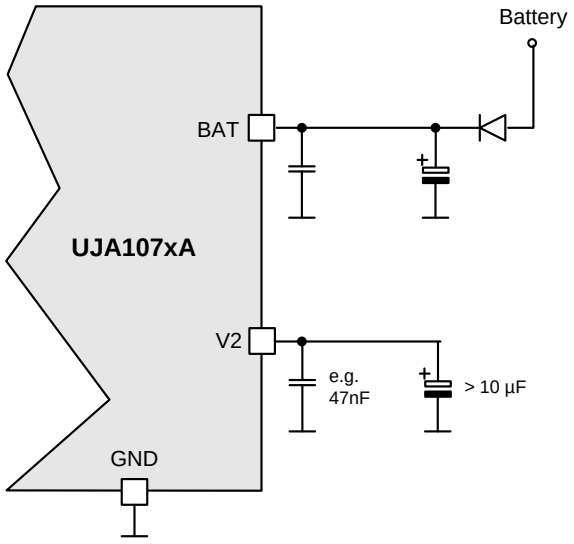


Fig 23. Typical V2 application

The V2 supply can also be used as an additional voltage supply for other hardware, e.g. additional transceivers etc. As the V2 supply delivers up to 120 mA current (V_{BAT} = 6 V to 28 V), even if V2 is used for supplying the internal CAN transceiver, additional current can be delivered for SBC external components. Indeed, it must be considered that the V2 regulator is only available in a dedicated Normal Mode (MC = 11) and any overload of V2 either caused by a CAN bus short circuit or by the additional hardware may result in a V2 undervoltage. The extra current that can be delivered from the V2 regulator for supplying additional hardware depends on the current consumption of the internal CAN Transceiver. If a CAN failure, e.g. CANH clamped to GND, is considered as normal use case, then in maximum the CAN transceiver requires 100 mA (short circuit peak current of the transceiver). Thus, at least 20mA are available for other ICs. In general, the V2 additional voltage supply depends on the system design (CAN termination, duty cycle of CAN signal etc.).

2.4.3.2 Configuration/Control of V2 regulator

The V2 supply can be activated and deactivated by the MC bits in the Mode Control Register. V2 activation is possible only in SBC Normal Mode and is independent from the CAN Transceiver state.

The current status of V2 can be read via the V2 status bit (V2S) of the Watchdog and Status Register in SBC's Normal Mode. The V2 status bit is always available in Normal Mode, regardless if V2 is supplied internally by the V2 regulator or externally (see section 2.4.3.3) by e.g. V1. In all other modes V2S is statically set to 1.

Additionally the SBC UJA107xA provides a V2 undervoltage warning interrupt that is triggered, if the output voltage drops below 90% of its nominal value. This interrupt can be enabled/disabled in the Interrupt Control Register via the V2UIE bit. The V2 undervoltage Interrupt Status can be read and cleared via the V2UI bit in the Interrupt Status Register.

2.4.3.3 External supply of the CAN Transceiver

In case a 5 V version of the UJA107xA is chosen, it is possible to save the separate buffer capacitors on V2 by supplying the internal CAN Transceiver also by V1 (see Fig 24). Furthermore, this supply option offers more flexibility for power distribution, e.g. by shifting power dissipation into the external PNP of V1 (see chapter 2.4.2).

To realize this approach, the Normal Mode with deactivated V2 must be chosen (MC = 2). Otherwise, there are two competing drivers for the CAN supply. The internal V2 undervoltage detection and V2 status monitoring also works in Normal Mode when V2 is deactivated (MC = 2) and indicate the status of the voltage at the V2 pin.

.

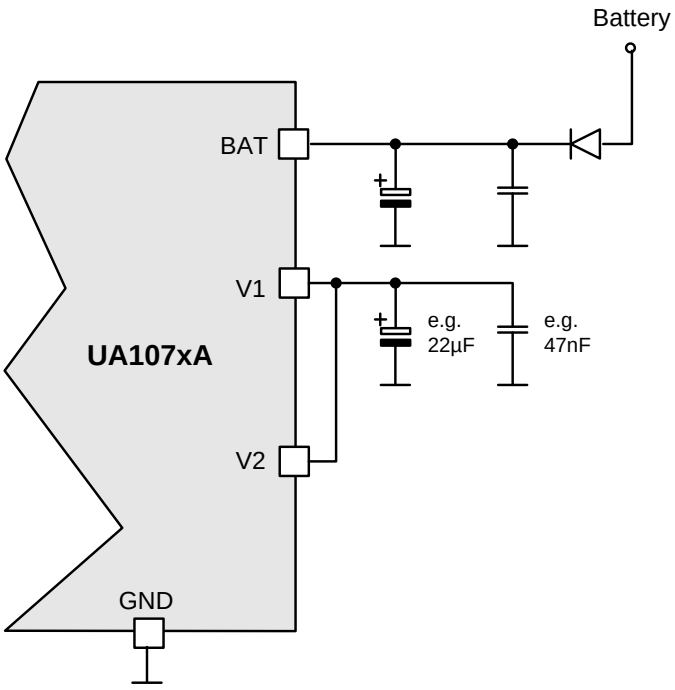


Fig 24. Example application for an externally supplied CAN physical layer

2.5 CAN Transceiver Interface

2.5.1 CAN Transceiver Overview

UJA1075A/76A/78A include a High-speed CAN Transceiver interface compliant to the ISO 11898-2 and ISO 11898-5 standards.

The CAN Transceiver is supplied by the internal V2 regulator or externally via the V2 pin. Fig 25 shows the operating voltage range of V2 for the CAN Transceiver. The CAN physical layer is fully specified between 4.75 V ≤ V_{V2} ≤ 5.25 V for a CAN termination between 45 Ω and 65 Ω.

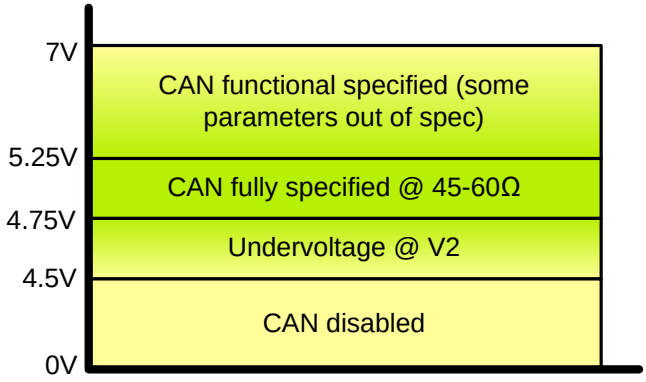


Fig 25. CAN Transceiver operating range

2.5.2 CAN Transceiver Operating Modes

The SBC CAN Transceiver supports different operating modes depending on the SBC mode and the CAN Standby Control bit (STBCC) in the Interrupt Control Register. The following table summarizes all CAN Transceiver operating modes:

Table 7. CAN Transceiver Operating Modes

	SBC is in Normal Mode (Mode Control Register: MC = 2|3)	**SBC is in Standby/Sleep Mode (Mode Control Register: MC = 0|1)**
Interrupt Control Register: STBCC = 0	*Active Mode:* CAN is enabled. The wake-up flag (visible on RXDC) is cleared regardless of V2 voltage level.	*Off Mode:* CAN is in Low-power Mode with disabled wake-up receiver. CAN bus pins terminated to ground. CAN wake-up interrupts can't be requested.
Interrupt Control Register: STBCC = 1	*Low-power Mode:* CAN is in Low-power Mode with enabled wake-up receiver regardless of the SBC mode (MC = 0|1|2|3), i.e. CAN wake-up interrupts can be requested	

Active Mode:

The CAN Transceiver is in Active Mode if

- The SBC is in Normal Mode (MC = 2|3)
- The transceiver has been enabled by clearing the according CAN Standby Control bit (STBCC = 0) and
- The voltage at pin V2 is above the V2 undervoltage detection threshold (regardless if CAN is supplied via V2 regulator or externally).

In CAN Transceiver Active Mode the CAN Transceiver is enabled and thus, data can be transmitted and received.

Low-power Mode:

It is possible to enter Low-power Mode regardless of the SBC mode by setting the CAN Standby Control bit (STBCC=1) in the Interrupt Control Register (ICR).

In low-power Mode the CAN Transceiver is able to receive a remote wake-up via CAN. In order to prevent unintended wake-ups the UJA107XA requires a wake-up sequence that consists of a recessive, dominant, recessive, dominant signal on the CAN-bus within the specified time $t_{to(wake)}$. Thereby each phase must be 0.5…3us ($t_{bus(rec)(min)}$, $t_{bus(dom)(min)}$). After receiving this wake-up pattern an interrupt is triggered if the SBC is in Standby or Normal. If the SBC is in Sleep Mode the device enters Standby Mode and enables the microcontroller supply V1.

Off Mode:

The CAN Transceiver is in Off Mode if the SBC is in Standby/Sleep Mode (MC = 0|1) and the CAN Standby Control bit has been cleared (STBCC=0).

In CAN Transceiver Off Mode the CAN Transceiver is completely powered down to save quiescent current. Hence the CAN bus pins are terminated to ground, the wake-up receiver is disabled and no wake-up interrupts can be requested.

2.5.3 CAN Transceiver RXDC/TXDC

The RXDC and TXDC wires are used for the serial communication between the CAN protocol controller and the UJA107xA (see Fig 26). These connections can optionally be applied with e.g. 1 kΩ series-resistors for filtering noise. But note that the series-resistors can have a negative impact on the loop delay. Please refer to the individual OEM hardware specification.

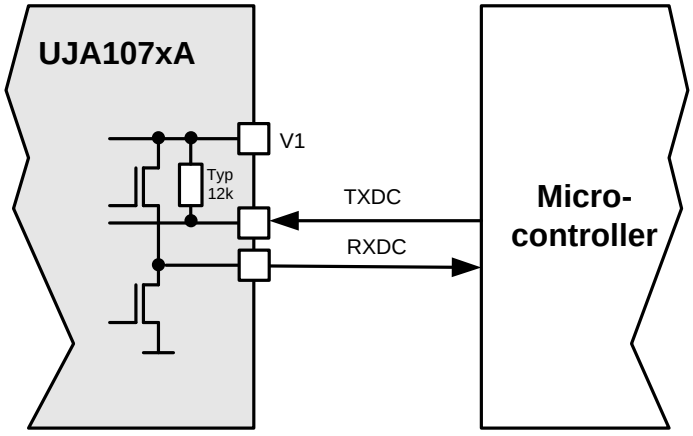


Fig 26. CAN Transceiver RXDC/TXDC Interface

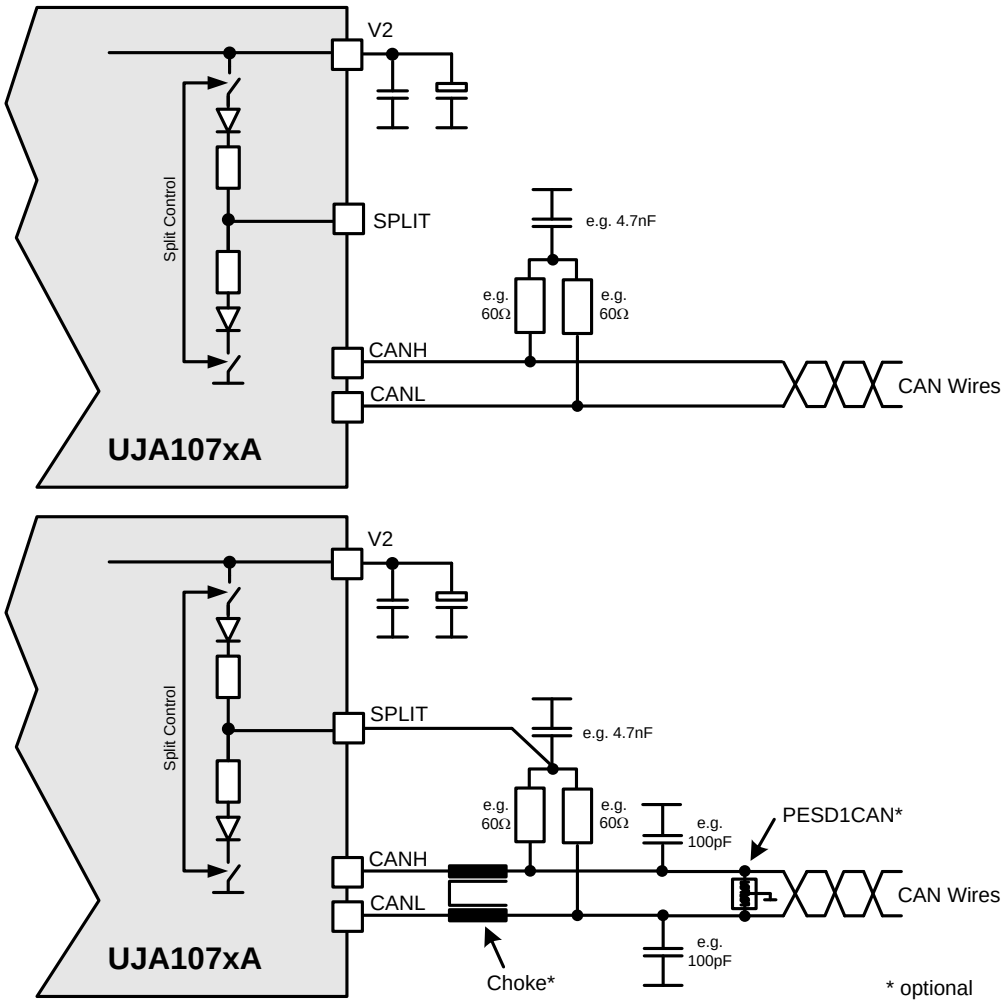
2.5.4 CAN Termination

In general the termination circuitry of the CAN Transceiver shall be designed according to the specification of the car manufacturer. The UJA107xA provides a SPLIT pin that delivers an output voltage of $V_2/2$. It is activated in CAN Transceiver Active Mode only. In CAN Transceiver Low-power and Off Mode, the SPLIT pin is floating.

The SPLIT pin in the high-speed CAN derivatives is intended for stabilizing the recessive level of the CAN bus. Therefore DC stabilization of the common mode voltage is achieved by simply connecting the SPLIT pin to the centre tap of the Split termination as shown in Fig 27.

The SPLIT pin can help to reduce electromagnetic emission of the CAN bus, especially if defect nodes cause unsymmetric loads in the CAN bus. However, the usage of the SPLIT depends on the carmaker's requirements. For details, please refer to the individual OEM specifications.

In case the SPLIT is not used, the SPLIT output pin can be left open (see Fig 27).



(1) Top: example for minimum circuitry needed for full CAN bus functionality

(2) Bottom: example for maximum circuitry with use of SPLIT pin, common mode choke and additional ESD protection

Fig 27. Typical Application of CAN Transceiver Block

A common mode choke reduces emission and improves immunity against common mode disturbances and ESD. If recommended by the carmaker a common mode choke can be used to reduce the impact of RF-interferences. Nevertheless, the performed IBEE EMC measurements [9] have proven that the UJA107xA meets the test conditions defined within the document "Hardware Requirements for LIN, CAN and FlexRay

Interfaces in Automotive Applications" [7] without a common mode choke. For details, please refer to the individual OEM specifications.

2.5.5 CAN ESD protection

The UJA107xA is designed to withstand ESD pulses up to 8 kV according to the Human Body Model (HBM, C = 100 pF, R = 1.5 kΩ) and at least 6 kV according to the IEC61000-4-2 (C = 150 pF, R = 330 Ω) at the bus pins CANH, CANL and pin SPLIT and thus, typically does not need further external measures. Nevertheless, if higher protection is required, external clamping circuits can be applied to the CANH and CANL line, e.g. PESD1CAN or PESD2CAN (see Fig 27). The IBEE ESD measurements were performed without external bus filters at CANH and CANL and confirmed an ESD robustness for pulses even higher than 6kV according to IEC61000-4-2 (C = 150 pF, R = 330 Ω). (For further information see [9])

2.6 LIN Transceiver Interface

2.6.1 LIN Transceiver Overview

The SBC derivatives UJA1075A/79A/78A include integrated LIN Transceiver Interface(s) that is/are directly supplied out of BAT. The slope timings and propagation delay symmetry is adapted to the LIN 2.1 standard in order to allow maximum bit rate tolerance in the system while offering excellent EMC performance.

2.6.2 LIN Transceiver Operating Modes

The SBC LIN Transceiver supports different operating modes depending on the SBC mode and the LIN Standby Control bit (STBCLx) in the Interrupt Control Register. In case of UJA1078A both LIN Transceivers can be controlled independently. The following table summarizes all LIN Transceiver operating modes:

Table 8. LIN Transceiver Operating Modes

	SBC is in Normal Mode (Mode Control Register: MC = 2|3)	**SBC is in Standby/Sleep Mode (Mode Control Register: MC = 0|1)**
Interrupt Control Register: STBCLx = 0	*Active Mode:* LINx is enabled. The wake-up flag (visible on RXDL1) is cleared regardless of BAT voltage level.	*Off Mode:* LINx is in Low-power Mode with disabled wake-up receiver. LIN bus pin is in high-resistive state. LINx wake-up interrupts can't be requested
Interrupt Control Register: STBCLx = 1	*Low-power Mode:* LINx is in Low-power Mode with enabled wake-up receiver, regardless of the SBC mode (MC = 0|1|2|3), i.e. LINx wake-up interrupts can be requested	

Active Mode:

In Active Mode the LIN Transceiver is enabled and can transmit and receive data. The LIN Transceiver is in Active Mode if

- The SBC is in Normal Mode (MC = 2|3),
- The transceiver has been enabled by clearing the LIN Standby Control bit (STBCLx=0) and
- The battery voltage is above the LIN undervoltage release threshold ($V_{uvr(LIN)}$).

Low-power Mode:

It is possible to enter Low-power Mode regardless of the SBC mode by setting the LIN Standby Control bit (STBLCx=1) in the Interrupt Control Register (ICR).

In Low-power Mode the LIN Transceiver is able to receive a remote wake-up via LIN if enabled. In order to prevent unintended wake-ups the UJA107xA requires a LOW time of at least $t_{bus(dom)(min)}$ followed by a rising edge on the LIN bus after the enabling of the LIN wake-up receiver.

Off Mode:

The LIN Transceiver is in Off Mode if the SBC is in Standby/Sleep Mode (MC = 0|1) and the according LIN Standby Control bit has been cleared (STBCLx=0).

In LIN Transceiver Off Mode the LIN Transceiver is completely powered down to save quiescent current. Hence, the LIN bus pin is in high-resistive state, the wake-up receiver is disabled and no wake-up interrupts can be requested.

2.6.3 LIN Transceiver RXDL/TXDL

The RXDL and TXDL wires are used for the serial communication between the LIN protocol controller and the UJA107xA (see Fig 28). These connections can optionally be applied with 1 kΩ series-resistors for filtering noise. For details, please refer to the individual OEM specifications.

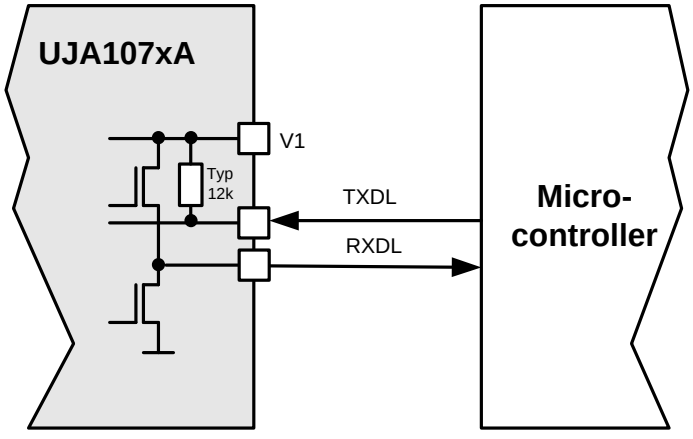
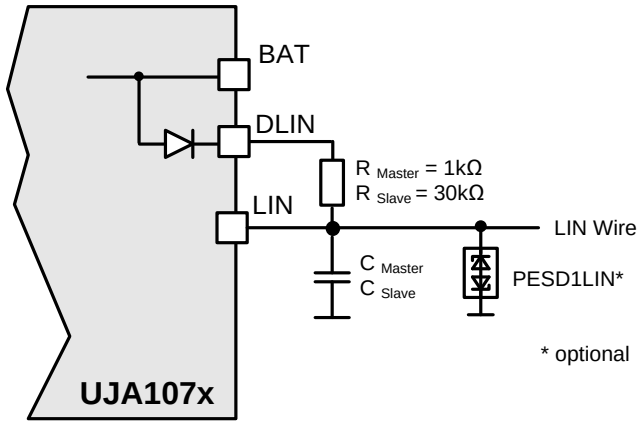


Fig 28. LIN Transceiver RXDL/TXDL Interface

2.6.4 LIN Termination

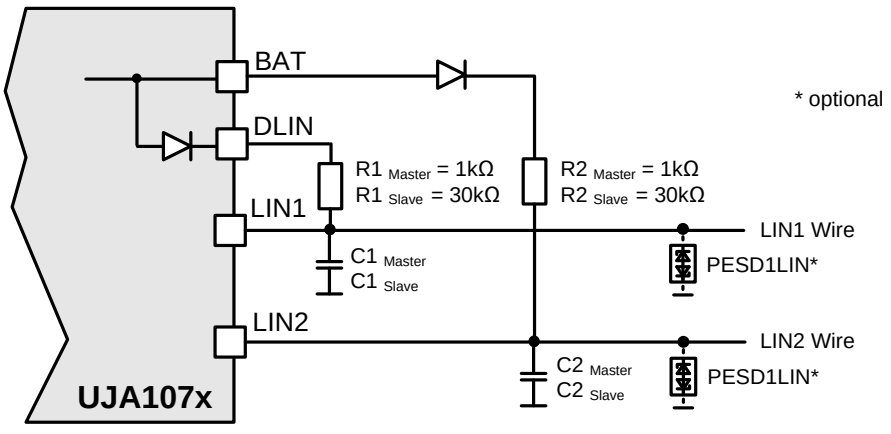
In general the termination circuitry of the LIN Transceiver shall be designed according to the specification of the car manufacturer. The UJA107xA family does not include an internal LIN termination resistor. Therefore, the termination resistor for LIN master applications (R_{Master} = 1 kΩ) as well as for LIN slave applications (R_{Slave} = 30 kΩ) needs to be connected externally (see Fig 29). The UJA107xA provides an integrated diode between the pins BAT and DLIN that can be used as LIN termination diode (see Fig 29).



Example circuitry with LIN standard conform bus capacitor and optional ESD protection

Fig 29. Typical Application of one LIN Transceiver Block

It is common practice to have one individual diode for each LIN channel. Therefore, if an UJA1078A is used, for the second LIN channel an external diode to battery must be used to terminate the LIN bus (see Fig 30).



Example circuitry with LIN standard conform bus capacitors and optional ESD protection

Fig 30. Typical Application of two LIN Transceiver Blocks

2.6.5 LIN Slope Configuration

The curve shaping of the LIN bus signal in normal slope mode is optimized for the maximum specified LIN transmission speed of 20 kBaud. Thus for low speed LIN or SAE2602 applications (≤10.4 kBit/s) an optimized curve shaping can be selected. For this the UJA107xA provides a special low slope mode with reduced slopes (see Fig 31). These reduced slopes result in further reduction of Electromagnetic Emission (EME).

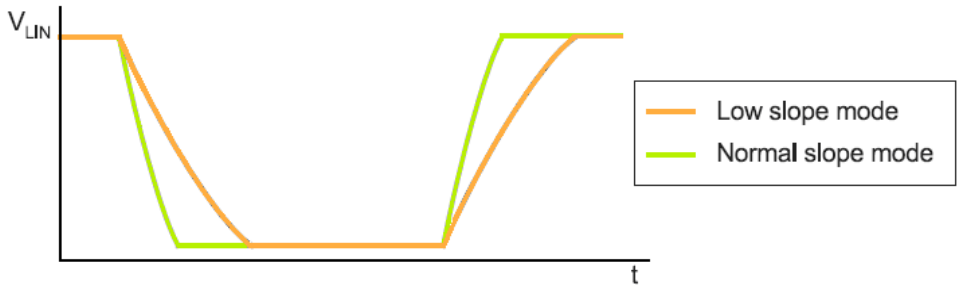


Fig 31. Reduced LIN bus slope in low slope Mode

To configure the LIN Transceiver for normal or low slope the LIN slope control (LSC) bit in the Mode Control Register (MCR) has to be set accordingly:

- Normal slope, 20 kBits: LSC = 0
- Low slope, 10.4 kBits: LSC = 1.

*For example to select the LIN normal slope mode, when the SBC is in Normal Mode with deactivated V2, the following SPI command has to be send to write MCR (X=don't cares; have to be set according to the application requirements): 001010XXX**0**XXXXXX$_b$.*

2.6.6 ISO 9141 (K-Line) Support

The Standard ISO 9141-2 "Road Vehicles – Diagnostic Systems – Part 2" [3] specifies the interchange of digital (diagnostic) information between on-board ECUs of road vehicles and a scan/test tool. The appropriate bus is the so-called "K-Line Bus".

Although the LIN physical layer [2] has been derived from the ISO 9141 [3] standard there are some differences such as shown in Table 9.

Table 9. Comparison ISO 9141 with LIN

Description	ISO 9141 [3]	LIN [2]	Compliance
Operating Voltage Range V_B	8 V to 16 V	7 V to 18 V	✓
Receiver High State	> 70% V_B	> 60% V_B	✓
Receiver Low State	< 30% V_B	< 40% V_B	✓
Temperature Range	0 °C to 50 °C	-40 °C to 125 °C	✓
Capacitance			
Diagnose Tester / LIN Master	< 2 nF	-	✓
ECU / LIN Slave	< 500 pF	< 250 pF	✓
Wiring	< 2 nF	< 6 nF	✓
Total	< 9.6 nF	< 10 nF	✓
Resistance			
Diagnose Tester / LIN Master	510 Ω	0.9 kΩ to 1.1 kΩ	✓
ECU / LIN Slave	> 100 kΩ	20 kΩ to 60 kΩ	✓ For reliable operation the overall pull-up of the network shall be above 450 Ω.
Timings			
Transmission Rate	10.4 kbit/s	1 kbit/s to 20 kbit/s	✓
Slew Rate / Slope Time	< 10 % T_{BIT} = 9.6 µs	duty cycle specification	The timing of the UJA107xA is according to the duty cycle specification of LIN, which is designed for communication speed of up to 20 kbit/s and results in better EMC compared to ISO 9141.

Although the LIN physical layer is not fully compliant to the ISO standard, the UJA107xA LIN transceiver will work in K-Line networks. The number of K-Line nodes could only be limited, if LIN transceivers with an integrated slave termination resistor (not existing in UJA107xA) are applied in the network. In a K-Line bus the overall network load is mainly caused by the Diagnose Tester (the master in a K-Line bus [3]), which is terminated with

a pull-up of R_{TESTER} = 510 Ω. But each LIN transceiver with integrated LIN slave resistor R_{SLAVE}, like e.g. the TJA1028 or TJA1021, will cause a decrease of the K-Line network resistance. The K-Line network resistance reduction can be calculated with following equation:

Minimum K-Line network load:

$$R_{K(BUS-BAT)\,\min} = \frac{R_{TESTER,\,\min} \times \frac{R_{SLAVE,\,\min}}{N}}{R_{TESTER,\,\min} + \frac{R_{SLAVE,\,\min}}{N}}$$

with

$R_{TESTER,min}$	minimum Diagnose Tester pull-up resistor
$R_{SLAVE,min}$	minimum LIN slave pull-up resistor
N	number of transceivers with integrated LIN slave resistor

Thus the maximum number of LIN transceivers (with integrated slave termination) in a K-Line bus is limited by the strength of the weakest bus driver. The UJA107xA is specified for the minimum network resistance of $R_{L(LIN-BAT)}$= 500 Ω [1] . Nevertheless the bus driver of the UJA107xA can drive a lower network resistance. The minimum bus resistance is $R_{K(BUS-BAT),min}$ = 450 Ω, which is derived from the minimum current limitation I_{BUS_LIM} [1] of bus driver.

Summary:

Though there are some deviations between the LIN and the ISO 9141 specification, the UJA107xA is able to support the K-Line bus from functional point of view. From a formal specification point of view, no LIN transceiver supports by 100% the original ISO 9141-2 specification [3].

2.6.7 LIN ESD protection

The UJA107xA is designed to withstand ESD pulses up to 8 kV according to the Human Body Model (HBM, C = 100 pF, R = 1.5 kΩ) and at least 6 kV according to the IEC61000-4-2 (C = 150 pF, R = 330 Ω) at the LIN bus pins and thus, typically does not need further external measures. Nevertheless, if higher protection is required, external clamping circuits can be applied to the LIN bus line, e.g. PESD1LIN (see Fig 29). The IBEE ESD measurements were performed without external bus filters at the LIN bus pins and confirmed an ESD robustness for pulses even higher than 6 kV according to IEC61000-4-2 (C = 150 pF, R = 330 Ω). (For further information see [9])

2.7 Local Wake-Up Interface

2.7.1 Functionality of WAKE pins

The UJA107xA offers 2 dedicated wake-up pins WAKE1 and WAKE2. In most applications these pins are connected to an external wake-up switch to GND. The wake-up pins can be configured individually to be either sampled continuously or sampled cyclically with two different periods. This configuration is done via the WBC bit in the Mode Control Register and the WSE bits in the Interrupt Control Register (see chapter 2.7.3 and 2.7.4). Furthermore, the edge sensitivity (falling, rising or both) of the wake-up pins can be configured separately via the WIC1 and WIC2 bits in the Interrupt Control Register (ICR). These bits can also be used to disable the WAKE pins completely.

A change of the wake sample configuration from continuous sampling (WBC = 1, WSE1 = 0, WSE2 = 0) to cyclic sampling (WSE1 = 1 or WSE2 = 1) and vice versa is possible at any point in time. In case cyclic sampling is enabled it is ensured that the WBIAS LOW time is always t_{WBIASL} (227µs - 278µs) and that the first sample point is within the configured cycle time t_{cy} (16ms or 64 ms; depending on the WBC setting).

When the wake-up is enabled (WICx != 0), a valid wake-up event will generate a wake-up interrupt. If the SBC is in Sleep Mode when the wake-up event occurs, it will wake up and automatically enter Standby Mode. If at least one WAKE interrupt is enabled, the status of the wake-up pins can be read via the WLS1 and WLS2 bits in the Watchdog and Status Register (WDSR). If no WAKE interrupt is enabled, the function of the WLS bits is deactivated and the value is set to “0”.

In case the WAKE pins are not used in the application, they can be left open or connected to ground. Furthermore, the according WICx bits should be set to 00 to completely disable the wake functionality.

If external wake-up circuitry is connected to a WAKE pin, it is recommended to let this circuitry be operational independent of the V1 and V2 output voltages of the SBC. Otherwise a wake-up from this circuitry via the WAKE pin may not be possible, while the SBC is in Sleep mode, i.e. while V1 and V2 are off.

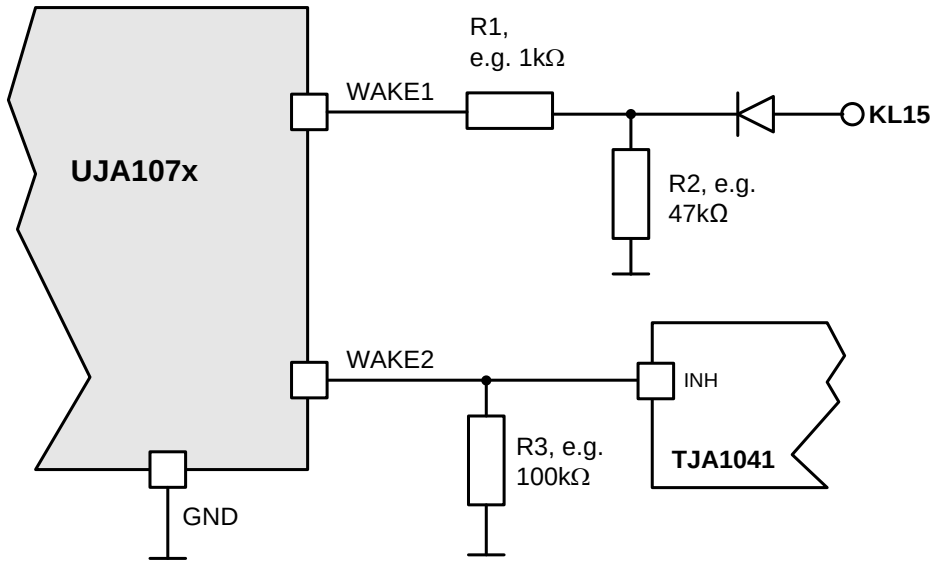
2.7.2 WAKE ESD protection

The WAKE pins of the UJA107xA are designed to withstand ESD pulses up to 8 kV according to the Human Body Model (HBM, C = 100 pF, R = 1.5 kΩ). A series resistor R1 (see Fig 32) is needed, when at least 6 kV according to IEC61000-4-2 (C = 150 pF, R = 330 Ω) is required. The IBEE ESD measurements were performed with R1=1 kΩ and confirmed an ESD robustness for pulses even higher than 6 kV according to IEC61000-4-2 (C = 150 pF, R = 330 Ω). (For further information see [9])

2.7.3 Continuously sampled wake-up mechanism

The WAKE pins can directly be connected to control signals (e.g. KL15 = battery signal that is turned on by the ignition key, or a transceiver INH signal) that shall trigger a SBC wake-up. To sample the wake-up pins continuously, the WSE1 and WSE2 bits in the Interrupt Control Register (ICR) must be configured to be 0. The edge sensitivity as well as the complete on-off control can individually be selected via the WIC1 and WIC2 bits in the Interrupt Control Register.

The following figure shows an example for a typical application of the direct wake-up mechanism. If a WAKE input is connected to KL15 an additional diode for reverse polarity protection is necessary. Furthermore, an external pull-down resistor is required to ensure that the WAKE pin level is LOW in case KL15 is deactivated. The recommended range for R1 is 100 Ω to 100 kΩ and for the pull-down resistors R2 and R3 is e.g. 47 kΩ and 100 kΩ. If the WAKE pin shall be robust against ESD pulses of 6 kV and higher according to IEC 61000-4-2 the series resistor R1 must be at least 1 kΩ. (see chapter 2.7.2)



KL15 = battery signal that is switched by the ignition key

Fig 32. Typical application of continuously sampling the WAKE pins

Below some example configurations for continuously sampled wake-up ports:

- *Both WAKE ports are sampled continuously; The wake interrupts shall be generated on both edges:*
  - *WSE1 = WSE2 = 0; WIC1 = WIC2 = 3*
  - *SPI command: ICR → 0100XXXX**1111**XX**00**$_b$*
- *Both WAKE ports are sampled continuously; WAKE1 interrupt shall be generated on falling edges, WAKE2 interrupt on both edges:*
  - *WSE1 = WSE2 = 0; WIC1 = 2; WIC2 = 3*
  - *SPI command: ICR → 0100XXXX**1011**XX**00**$_b$*
- *Only WAKE1 port is sampled continuously; WAKE2 pin is off; WAKE1 interrupt shall be generated on rising edges*
  - *WSE1 = 0; WIC1 = 1; WIC2 = 0*
  - *SPI command: ICR → 0100XXXX**0100**XX**0**X$_b$*

2.7.4 Cyclic sampled wake-up mechanism for power saving

A significant reduction of quiescent current can be achieved when the cyclic biasing and sampling feature of the UJA107xA is used to detect status changes at the WAKE inputs. This is realized by synchronizing the sampling of the WAKE inputs to the WBIAS signal und use the cyclic WBIAS signal for supplying the switches.

The WBIAS pin is a battery-related, active-LOW, open-drain output pin that controls the bias transistor of the switches. The synchronization between WBIAS pin and the WAKE pins is enabled via the WSE1 or WSE2 bits in the Interrupt Control Register. If at least one WSEx bit is set, cyclic biasing via the WBIAS signal is activated. The bias and sampling period (16 ms or 64 ms) is configurable via the WBC bit in the Mode Control Register. The sampling time (WBIAS low) is about 250 µs (t_{WBIASL}) and sampling will be performed at the end of the sampling time (on the rising edge of WBIAS).

Fig 33 shows a typical application for cyclic sampled wake-up inputs. WAKE1 and WAKE2 are connected to switches to GND, with pull-up resistors that are supplied by the bias transistor under control of WBIAS.

Note that for selection of the values of the pull-up resistors in the figure below, it should be taken into account that optional capacitors at the switch signals need to be charged in time. In order to detect an open contact correctly, the charge time of all capacitive loads connected to the pin WAKE has to be considered. That means, the upper WAKE threshold voltage has to be passed in order to start the wake-up filter time (t_{wake}), which must be elapsed before the WBIAS signal is switched off again (Sample of WAKEx).

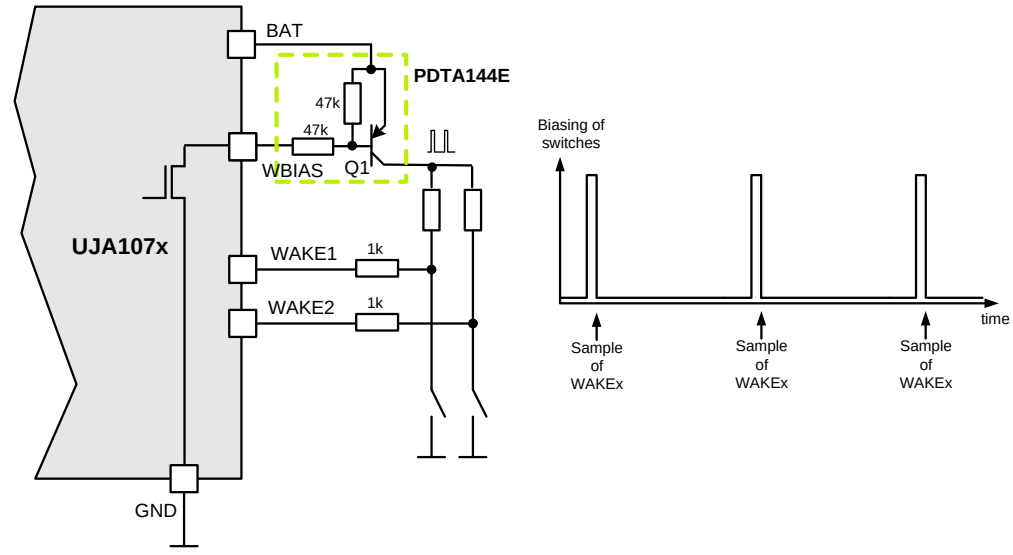


Fig 33. Typical application of cyclic sampling of switches

The WBIAS can also be statically activated to continuously supply the switches (WSE1 = WSE2 = 0 and WBC = 1). In that case the behavior and the configuration of the WAKE pins is conform to that of continuously sampled wake-up pins (see 2.7.3).

Below some example configurations for cyclic sampled wake-up ports:

- *Both WAKE ports are sampled cyclically with 16ms; The wake interrupts shall be generated on both edges:*
  - *WSE1 = WSE2 = 1; WIC1 = WIC2 = 3; WBC = 0*
  - *SPI commands:* *ICR → 0100XXXX**1111**XX**11**$_b$*
    *MCR → 0010XXXXXX**0**XXXXX$_b$*
- *Both WAKE ports are sampled cyclically with 64ms; Wake interrupt shall be generated on rising edge for WAKE1, on falling edge for WAKE2:*
  - *WSE1 = WSE2 = 1; WIC1 = 1, WIC2 = 2; WBC = 1*
  - *SPI commands:* *ICR → 0100XXXX**0110**XX**11**$_b$*
    *MCR → 0010XXXXXX**1**XXXXX$_b$*
- *WAKE1 is sampled cyclically with 16ms; WAKE1 interrupt shall be generated on falling edge; WAKE2 is not connected to the WBIAS transistor and thus, shall be sampled continuously; WAKE2 interrupt shall be generated on both edges:*
  - *WSE1 = 1; WSE2 = 0; WIC1 = 2; WIC2 = 3; WBC = 0*
  - *SPI commands:* *ICR → 0100XXXX**1011**XX**10**$_b$*
    *MCR → 0010XXXXXX**0**XXXXX$_b$*

2.7.4.1 How to reduce the effective on-time of WBIAS

The WBIAS on-time of the UJA107xA is about 250µs (t_{WBIASL}). Normally the WBIAS output directly controls the bias transistor (Q1 in Fig 33) and the on-time of this transistor equals the on-time of WBIAS. When the switch contacts are closed, the bias current is flowing while the bias transistor is turned on. This current contributes to the average power consumption of the module. In order to further decrease the current consumption during cyclic sampling of the Wake inputs, the turn-on of the bias transistor can be delayed by external circuitry, which has the same effect like a reduction of the WBIAS on-time would have. This can be useful especially when a high number of switches is connected.

Fig 34 below shows the Wake signal with normal and with delayed turn-on of the bias transistor. With normal turn-on of the bias transistor, the wake signal rises almost immediately after the falling edge of the WBIAS signal. The rise time is mainly determined by the bias resistor and the capacitive load at the switch. When the rise time is fast enough, the WBIAS signal may contain an on-time overhead, as the wake level is always sampled at the end of the WBIAS on-time. The on-time overhead can be calculated as follows:

on-time overhead = WBIAS on-time - Wake rise time - Wake filter time

With a delayed turn-on of the bias transistor the on-time overhead can be minimized:

on-time overhead = WBIAS on-time - Wake rise time - Wake filter time – delay time

The delay time in Fig 34 (bottom) is achieved by delayed enabling of the external bias transistor Q1.

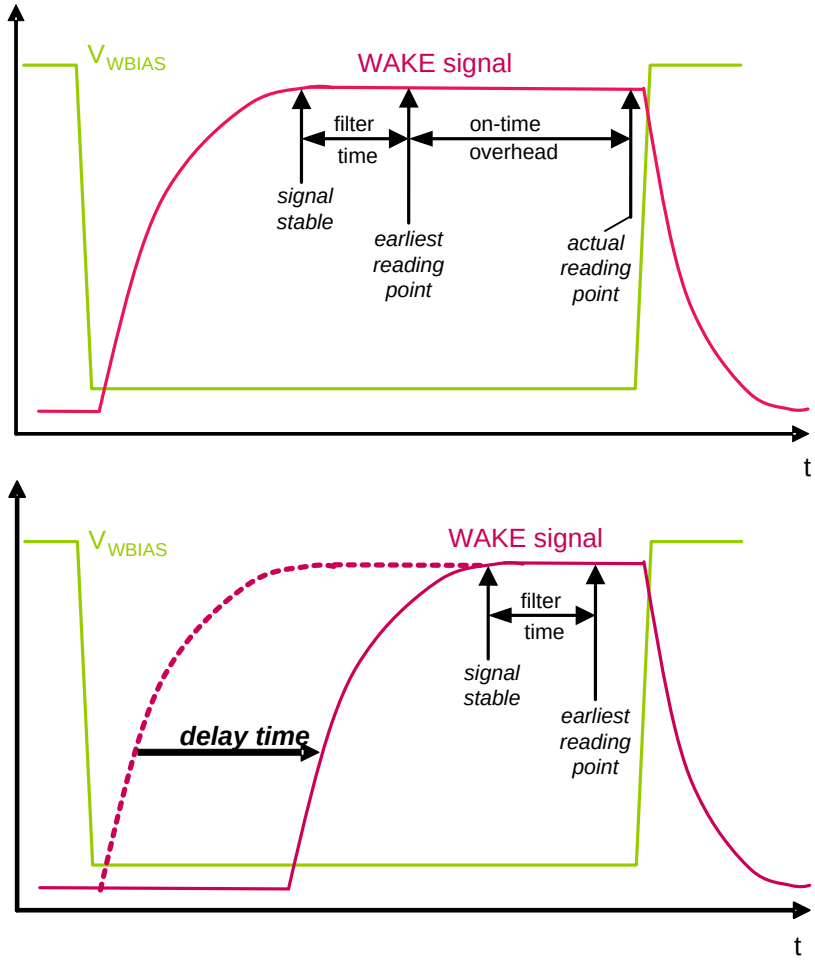


Fig 34. Normal (top) vs. delayed (bottom) turn-on of the bias transistor.

Fig 35 shows the delay circuit for reduction of the effective WBIAS on-time. Q1 is the bias transistor. This time the type of Q1 has been chosen so that up to 24 switches can be connected with 1 kΩ bias resistors. Q2 represents a current amplifier, so that the R2/C1 timer can run with low currents and thus with low capacitor values. The double diode D1 serves as voltage reference, which makes the charging time of C1 independent

of the battery voltage and compensates the temperature drift of the basis-emitter diode of Q2. Therefore D1 and Q2 should be placed close together on the PCB. The values of C1 and R2 in the figure below will result in a delay of typ. 150 µs. Therefore the shown circuit will generate a WBIAS on-time of approx. 250 µs-150 µs = 100 µs. The circuitry itself consumes almost no power, so that the power savings by shortening the pulse length can be fully exploited. The accuracy of the delay depends mainly on the accuracy of the capacitor. However, if no load is connected to the circuit (all switches open), the delay is much shorter than expected. This does not matter because there is no current to save either. As soon as one or more switches are closed (i.e. the bias current gets in the order of 10 mA or more), the delayed turn-on of the bias transistor gets visible.

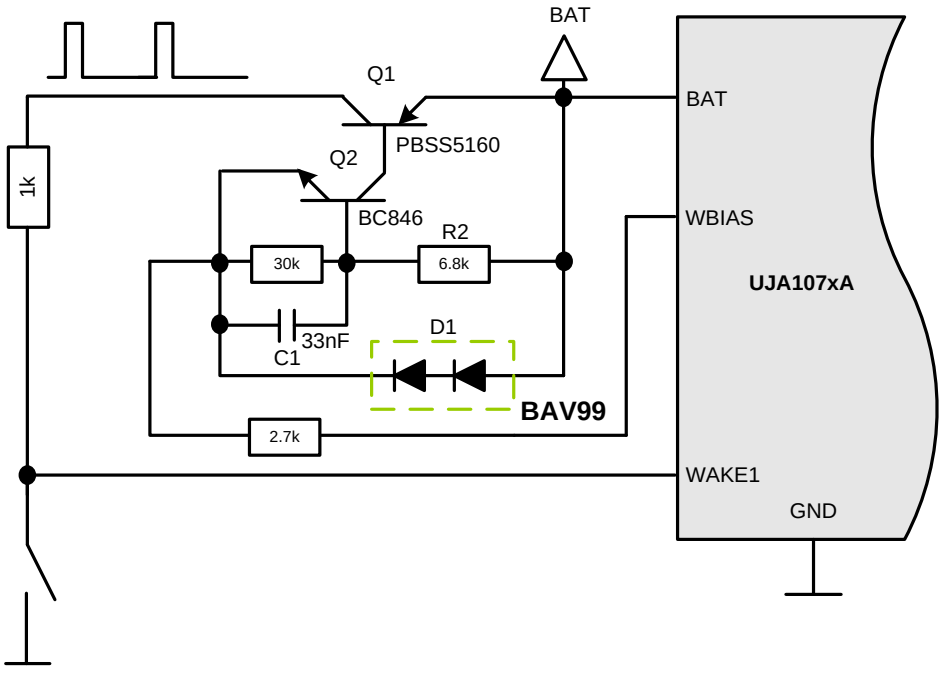


Fig 35. Circuit to delay the turn-on of the bias transistor Q1, which reduces the effective WBIAS on-time

The turn-on delay can easily be changed by appropriate selection of R2 and C1. Fig 36 below shows the nominal delay time of the circuit with respect to R2 and C1. The resulting nominal on-time can be calculated as:

on-time = 250 µs - nominal delay time.

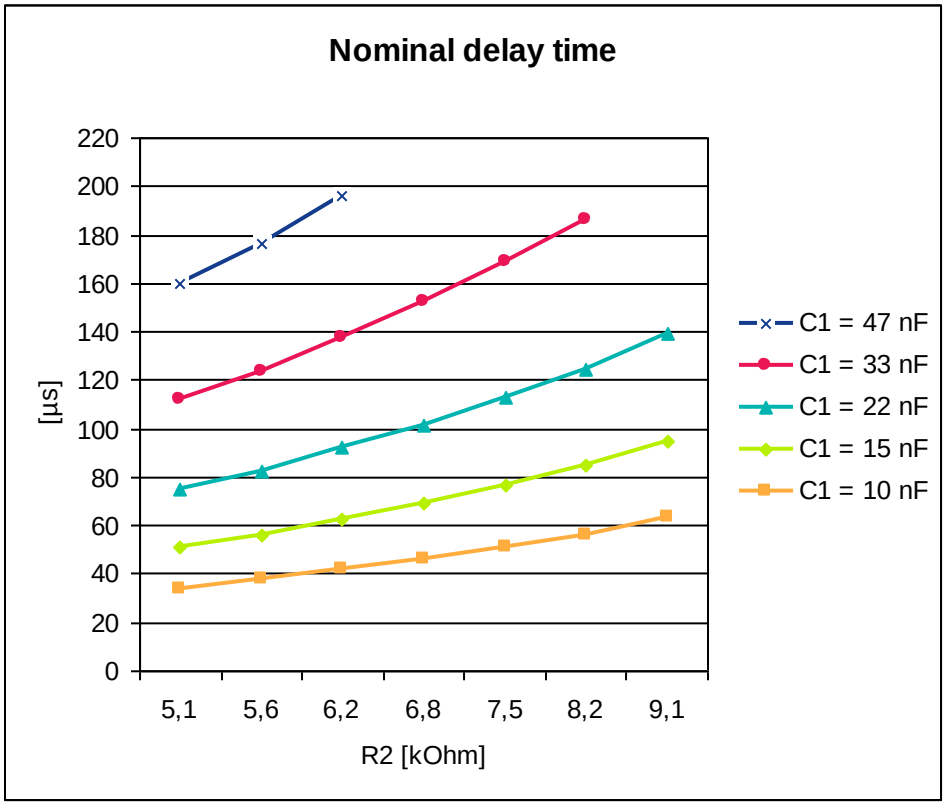


Fig 36. The nominal turn-on delay time of the bias transistor Q1 as a function of C1 and R2.

2.7.5 UJA107xA wake-up port extension with PCA9703

In applications with the requirements for more than 2 local wake-up ports the PCA9703 can be used to increase the number of wake-up ports by 16 or multiples of 16 (with multiple PCA9703 ICs). The PCA9703 is a low power 18V tolerant SPI General Purpose Input (GPI) shift register designed to monitor the status of switch inputs. It generates an interrupt when one or more of the switch inputs change state but allows selected inputs to not generate interrupts using the interrupt masking feature. The input level is recognized as a HIGH when it is greater than $0.8 \times V_{DD}$ and as a LOW when it is less than $0.55 \times V_{DD}$ (minimum LOW threshold of 2.5 V at 5 V node). For more information on the PCA9703 refer to the data sheet [8] .

The following sub chapters will introduce the minimum application circuitry for the interoperation of the UJA107xA and PCA9703, allowing to use the cyclic sampled wake mechanism of the UJA107xA in conjunction with the PCA9703, for minimized power consumption in standby or sleep mode.

2.7.5.1 PCA9703 configuration for standby applications

This section shows the usage of the PCA9703 in applications that operate in Normal or Standby Mode only (V1 always enabled). In this case the PCA9703 can be supplied out of V1, see Fig 37. The transistor Q3, connected to the INTN and INT_EN pin of the PCA9703, will enable the interrupt output only while the switches connected to the PCA9703 are supplied, i.e. while WBIAS is low. This will further reduce the current consumption of the PCA9703, because the highest current consumption of the PCA9703 is with enabled interrupts. Note that while the INT_EN pin is low, the inputs of the PCA9703 are disabled and a SPI read-out reflects the input signal status right before the INT_EN went low.

When the ECU shall enter standby mode, the cyclic sampled wake-up mechanism (see section 2.7.4) has to be enabled within the UJA107xA. Moreover, the WAKE1 interrupt is configured on falling edge polarity because only on falling edges of the INTN pin of the PCA9703 a microcontroller service is required. Therefore, at least bits WSE1 = 1 and WIC1 = 2 must be set in the Interrupt Control Register (ICR) of the UJA107xA.

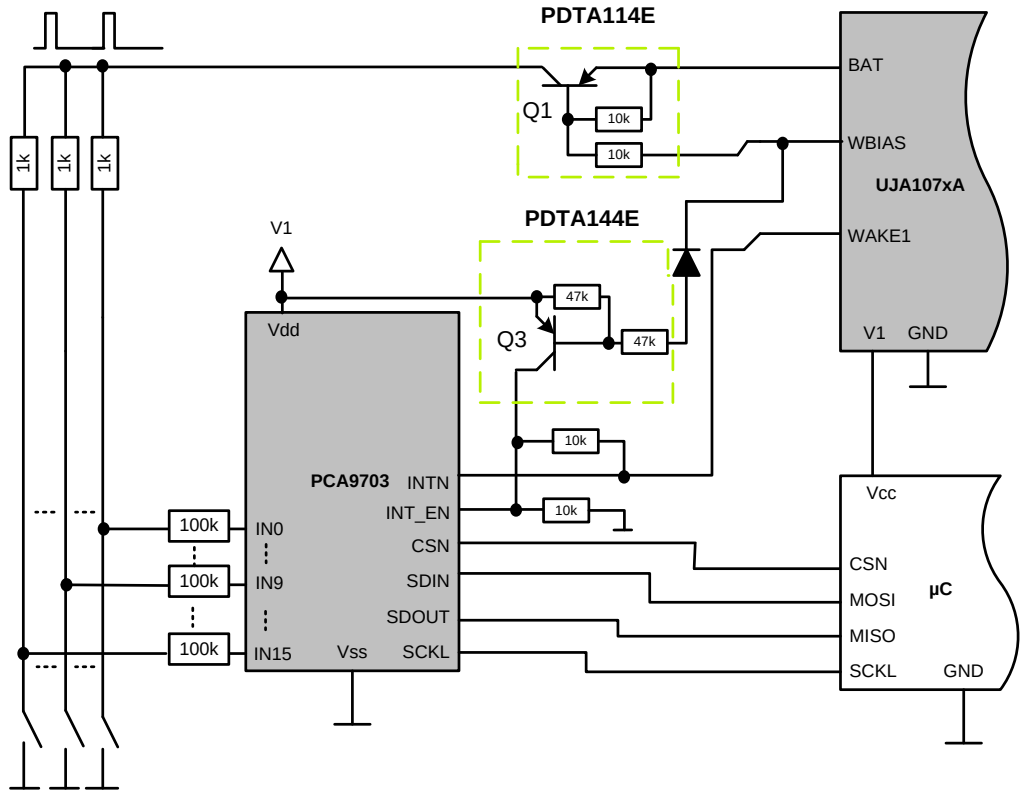


Fig 37. PCA9703 configuration for standby applications

After detection of a PCA9703 interrupt by the UJA107xA, the microcontroller will be woken-up with a WAKE1 interrupt by the UJA107xA, thus indicating that a read-out of the PCA9703 is required. Each interrupt of the PCA9703 must be serviced with a SPI access before the next sampling of the switches takes place (service time < 16ms (WBC=0) or 64ms (WBC = 1), respectively). Otherwise the interrupt source may disappear because with the next WBIAS pulse the INT_EN signal is activated again and thus the PCA9703 inputs are ready to read the current, new status of the switches.

The interrupt detection in the PCA9703 is done by comparison of the previously latched values of the last SPI transfer with the current values at the input pins. Hence, each SPI access will update the basis for the comparison.

SPI write accesses to the PCA9703 will always set the interrupt enable mask of the device. The read-out data of each SPI access represents the actual levels of all inputs, independently of this mask. Two things are important when writing SPI commands to the PCA9703.

- Always write the correct interrupt enable mask
- Compare the read back data with previous read back data on those bit positions that are marked with a "1" in the enable mask, in order to identify those signal(s) that have caused the interrupt

The software handling of the PCA9703 after wake-up depends on the needs of the application. Two cases are discussed here:

- the application permanently samples the switches in normal operation mode, or
- the switches are cyclically sampled in normal operation mode.

The difference is related to the configuration of the WAKE pin within the UJA107xA.

The easiest case is the normal operation with permanently sampled switches. In this application mode the software can perform the following actions. After a valid wake-up has been detected, the sampling of the switches is permanently enabled by setting WSE1=0, WSE2 = 0 and WBC = 1. The WBC change can be done with the same command that performs the transition from Standby to Normal Mode of the UJA107xA.

When entering Standby Mode again, the cyclic sampling must be re-enabled by setting the related WSEn bit to 1.

When the biasing is permanently enabled, it is also possible to poll the input of the PCA9703 by cyclically writing SPI commands to the PCA9703 in Normal Mode. On the other hand, each level change of the inputs of the PCA9703 can immediately trigger an interrupt within the UJA107xA.

In case that the switches connected to the PCA9703 shall be cyclically sampled also in Normal Operation Mode, the software flow slightly differs from the previous one. When an interrupt has been detected and the wake-up source is determined, the UJA107xA enters Normal Mode and cyclic sampling of the switches is still enabled. Therefore, there is no need to change the configuration of WSEn and WBC. However, after the WIn (WAKEn interrupt) has been cleared, the software has to check if the next sample of the INTN pin of the PCA9703 is high, because the edge detector of the WAKE pin needs at least one sample with high level before it can be triggered by a new sample with low level again. If INTN is still LOW, the software again has to perform an SPI access to the PCA9703, because there is apparently still an interrupt pending that has to be cleared in

order to release the INTN line. This procedure must be repeated until the software detects a high level on the INTN pin of the PCA9703 (i.e. at the WAKE pin to which the INTN line is connected).

Note that in this configuration it is not possible to poll switches connected to the PCA9703 by software, because such polling cannot be synchronized properly with the WBIAS pulse. Changes of the input level of the PCA9703 can only be detected via the UJA107xA WAKE interrupt.

2.7.5.2 PCA9703 configuration for sleep applications

This chapter shows the usage of the PCA9703 in applications, which operate in Sleep Mode (V1 can be disabled). The circuit in Fig 38 below also works in Standby Mode, but requires more external components than the previously discussed proposal. The reason is that the PCA9703 needs its own supply voltage and cannot be connected to the V1 supply of the UJA107xA. The PCA9703 supply is realized by a capacitor connected to the emitter of the combined transistor / Zener diode Q4 of type PVR100AD-B5V0. This component charges the capacitor with each sampling pulse. A capacitor of 470nF is sufficient to keep the PCA9703 functional during sleep mode of the application. The worst case conditions for calculation the capacitor size are:

- The UJA107xA sampling period (16 ms / 64 ms)
- The max. current of the PCA9703 (1 µA)
- The max drop of the capacitor voltage during discharge (150 mV is chosen because the minimum output voltage of the PVR100AD is 4.65 V and the min operating voltage of the PCA9703 is 4.5 V. 4.65 V - 4.5 V = 150 mV)

$$C = \frac{I \times t}{\Delta U} = \frac{1\mu A \times 64ms}{150mV} = 426nF$$

The transistor connected to the INTN and INT_EN pin of the PCA9703 will only enable the interrupt output when also the switches connected to the PCA9703 are supplied. This will further reduce the current consumption of the PCA9703 because the highest current consumption of the PCA9703 is with enabled interrupts.

Another transistor is required to prevent the reverse supply of the microcontroller by the PCA9703 during Sleep Mode. Therefore the PDTC144T is controlled by the RSTN pin of the UJA107xA, because the RSTN is always low in sleep mode.

In the configuration below the sampling of the switches is enabled within the UJA107xA. Moreover, the WAKE1 interrupt is configured on falling edge polarity because only on falling edges of the INTN pin of the PCA9703 a microcontroller service is required. Therefore, only bits WSE1=1 and WIC1=2 must be set in the Interrupt Control Register (ICR) of the UJA107xA.

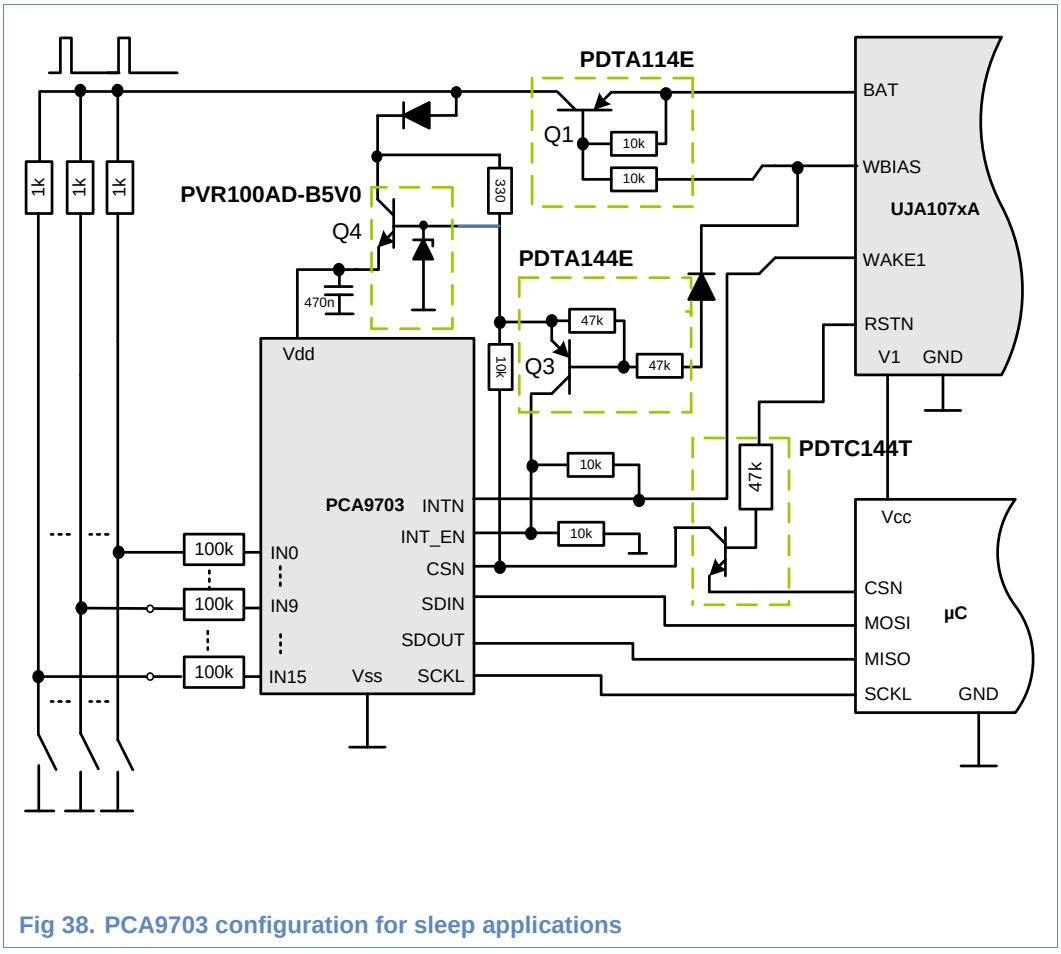


Fig 38. PCA9703 configuration for sleep applications

When a pin level change on the PCA9703 will take place during Sleep Mode of the application, the UJA107xA will detect the level change on the INTN pin of the PCA9703 and will wake-up the application. The wake-up source detection will only work when the following conditions are satisfied:

1. The levels of the inputs before entering sleep mode have been stored in a non-volatile memory to compare these values against the actual read back values of the PCA9703 after wake-up.
2. The root cause of the wake-up must still be present.

Otherwise the wake-up source is not detectable.

2.8 Failsafe Interface

The UJA107xA provides different possibilities to ensure a correct system behavior also at system malfunctions. This functionality is supported by the ENable and LIMP Interface.

Table 10 compares the main features and characteristics of both UJA107xA pins and summarizes the area of application. Additional information is given in the following sub chapters.

Table 10. Comparison of ENable and LIMP Interface

	ENable Interface	LIMP Interface
Application area	Used for automatic deactivation of hardware in case of a system failure	Used to activate emergency hardware in case of a serious system malfunction (by application or automatically upon a system reset)
Pin characteristic	V1-related push-pull output pin Active HIGH	Battery-related open-drain output pin Active LOW
Can be activated in…	UJA107xA Normal Mode only	UJA107xA Standby, Normal and Overtemp Mode
Is controlled by…	ENC bit in Mode Control Register and UJA107xA mode (only if ENC is set and UJA107xA is in Normal mode the EN pin is HIGH)	LHC in Mode Control Register for application activation LHWC in Mode Control Register for automatic activation upon a system reset
Output pin is activated when…	- SBC sets the ENC bit and SBC is in Normal Mode - SBC enters Normal Mode and ENC bit is set	- Software sets LHC bit - Two consecutive system resets without clearing LHWC bit in-between occur - V1 clamped LOW for longer than $t_{det(CL)L}$ though driven HIGH by the SBC (only supported for WD versions with WDOFF pin connected to ground) - RSTN clamped LOW for longer than $t_{det(CL)L}$ though driven HIGH by the SBC (only supported for WD versions with WDOFF pin connected to ground) - SBC enters Overtemp Mode
Output pin is deactivated when…	- Software clears the ENC bit - SBC leaves Normal Mode - Power-on	- Software clears LHC bit - Power-on

2.8.1 ENable Interface

To prevent safety relevant application hardware from being unintentionally activated, the UJA107xA provides a V1-related push-pull enable output pin (EN) that can be controlled by the application software via the SPI interface.

2.8.1.1 Scope of application

The EN pin can be used for fast and automatic deactivation of hardware in case of a system failure. The EN output function is linked to the UJA107xA mode, which allows disabling e.g. safety relevant hardware automatically whenever the UJA107xA is not in Normal Mode. Therefore the EN pin can only be set HIGH, if the SBC is in Normal Mode. The EN output is immediately set LOW as soon as RSTN is forced LOW or Overtemp Mode or any Low-power Mode (Standby/Sleep) Mode is entered. Fig 39 shows a typical application of the ENable output.

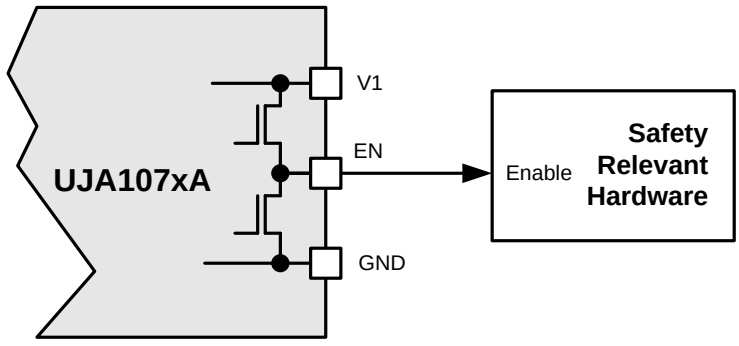


Fig 39. Typical application of the ENable output

The EN pin can also be used for a two-path-control of e.g. an output driver as shown in Fig 40. The output driver is only active if both, the microcontroller port pin and the EN pin are set to HIGH.

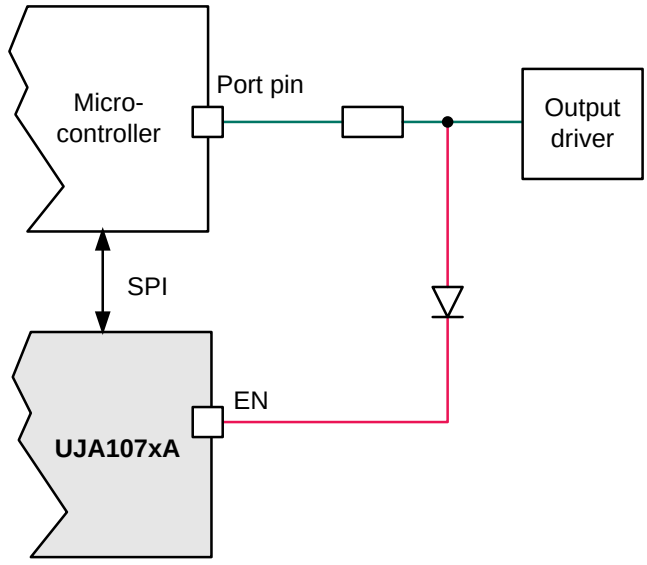


Fig 40. Two-path-control of an output driver

2.8.1.2 ENable output configuration

In Normal Mode the EN pin is directly controlled by the ENC bit in the Mode Control Register (MCR), else it is driven LOW.

2.8.2 LIMP Home Interface

To activate emergency hardware in case of a serious system malfunction, the UJA107xA provides a battery-related, active-LOW, open-drain output pin (LIMP) that is automatically activated and can also be controlled by the application software via the SPI interface.

2.8.2.1 Scope of application

The LIMP Home output can be used to activate application-specific ‘Limp-Home’ hardware in the event of a serious system malfunction. Detectable failure conditions are SBC overtemperature events, wrong watchdog service, V1 or RSTN clamped LOW (only supported for types with watchdog and WDOFF connected to ground), V1 failures and user-initiated or external reset events.

It can be selected if the LIMP pin shall be activated immediately after the first or only after repeated failure detection (see chapter 2.8.2.2).

Fig 41 shows a typical application of the LIMP output.

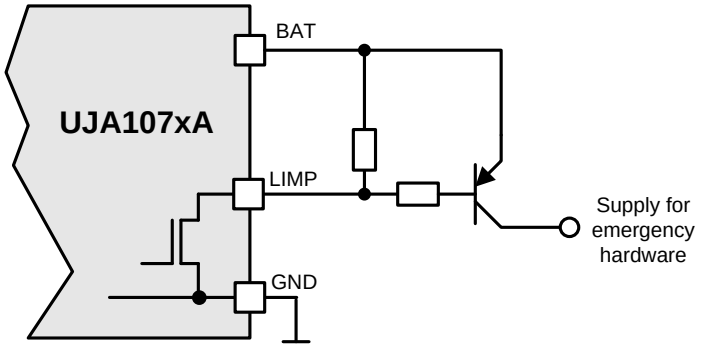


Fig 41. Typical application of LIMP output

2.8.2.2 LIMP Home configuration

There are two bits in the Mode Control Register (MCR) that allow control of the LIMP Home feature: Limp Home Warning Control (LHWC) and Limp Home Control (LHC).

Activation of LIMP by the application:

The application can activate the LIMP output at any time by setting LHC.

Automatic activation of LIMP upon each system reset:

When LIMP shall be activated immediately upon any system reset, the application has to keep the LHWC bit set, which is its default state. Note that the power-on reset of the SBC does not activate LIMP.

Automatic activation of LIMP only after serious system failure:

A single system reset is not regarded as a serious system failure. However, two consecutive system resets without meaningful microcontroller interaction in-between is regarded as a serious system failure. If the LIMP output shall only be activated in such a case, the application has to clear the LHWC bit after each system reset. This way the next system reset does not activate LIMP. However, when e.g. the microcontroller is no longer able to clear this bit, the LIMP output will be activated with the next system reset (e.g. because of watchdog overflow).

Another serious system failure is present, when the SBC enters Overtemp Mode, the V1 pin is clamped LOW or the RSTN pin is clamped LOW. These system failures set the LHC bit automatically and thus activate the LIMP output immediately or after the clamping detection time ($t_{det(CL)L}$), respectively regardless of LHWC setting. Please note, that failure detection of “V1 and RSTN pin is clamped LOW” is only available if an UJA107xA type with watchdog is used and the WDOFF pin is connected to ground.

2.9 Watchdog Interface

2.9.1 Watchdog overview

The UJA107xA contains a programmable watchdog with an independent clock source. This watchdog can be operated in Timeout Mode, Window Mode or can safely be disabled in Low-power Mode. Furthermore, the UJA107xA provides the possibility to completely disable the watchdog by hardware if not needed by the application or during software debugging (see chapter 2.9.2).

Both, in Normal and in Standby Mode the SBC supports a relaxed and a normal watchdog triggering depending on the WMC bit setting. Table 11 summarizes all supported watchdog modes.

- In Normal Mode the microcontroller has unlimited access to all system functions including CAN and LIN bus networking, a strict observation of the system behavior by the watchdog is required. Therefore in Normal Mode the watchdog cannot be deactivated by SW and works in Timeout or Window Mode.
- In Low-power Mode (Standby) the bus transceivers are deactivated for reducing power consumption and the EN output is low, disabling e.g. safety-relevant hardware. Hence the watchdog can be disabled or works in Timeout Mode.

Table 11. Supported watchdog modes

	WMC	Normal Mode (MC = 2|3)	Standby Mode (MC = 0)
Normal watchdog triggering	0	watchdog in Window Mode	watchdog in Timeout Mode
Relaxed watchdog triggering	1	watchdog in Timeout Mode	watchdog off

The SBC supports 8 watchdog periods: 8ms, 16ms, 32ms, 64ms, 128ms (default), 256ms, 1024ms and 4096ms. The default watchdog period is 128ms.

2.9.2 Functionality of the WDOFF pin

For easy software development the UJA107xA supports the possibility to disable the watchdog completely by use of the WDOFF pin. The WDOFF pin is a V1 related digital input pin that can be connected to GND or V1. If the pin is set to HIGH by connecting it to V1, the watchdog is completely disabled, overruling any software watchdog configurations. Otherwise (WDOFF connected to GND) the watchdog configuration is determined by the WMC bit, as described above. A change of the WDOFF pin status always causes a system reset in Normal or Standby Mode. The internal circuitry either drives the WDOFF pin

- LOW, if it is has been connected to GND for at least the watchdog off filter time (t_{fltr})or
- HIGH, if it has been connected to V1 for at least the watchdog off filter time (t_{fltr}).

The figure below illustrates WDOFF circuitry in case the watchdog shall be disabled during software development (left - WDOFF connected to V1), but enabled for series production (right – WDOFF connected to GND). The value for the external pull-down and pull-up resistor respectively must be smaller than the internal pull-up/pull-down resistance R_{pupd} (5 kΩ – 20 kΩ), e.g. 1 kΩ.

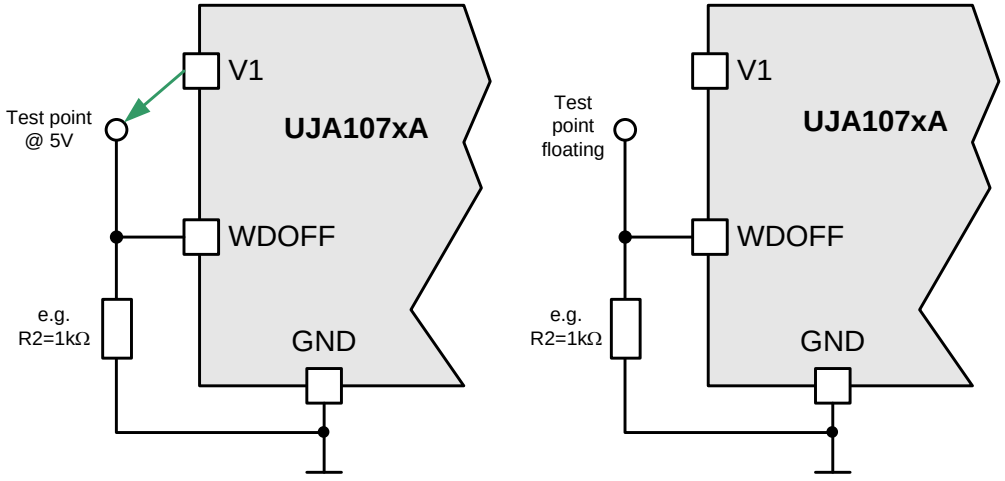


Fig 42. WDOFF connection for disabled (left) and enabled watchdog (right)

In case no watchdog is available in the UJA107xA device, the WDOFF pin should be connected to ground.

2.9.3 Configuration of Watchdog Mode and Period

The UJA107xA supports different watchdog modes and 8 different watchdog periods: 8 ms, 16 ms, 32 ms, 64 ms, 128 ms (default), 256 ms, 1024 ms and 4096 ms. The watchdog is controlled and configured by read or write accesses to the Watchdog and Status Register (WDSR). The watchdog mode is controlled by the Watchdog Mode Control bit (WMC) and the watchdog period by the Nominal Watchdog Period bits (NWP).

As every valid write access to the Watchdog and Status Register is interpreted as a watchdog trigger, the watchdog configuration is only allowed during an open watchdog window. Hence, changes to the watchdog must be coupled to the watchdog trigger routine. Beside the watchdog configuration bits the Watchdog and Status Register also contains a Software Reset bit (SWR: Bit 7) and read-only status bits. Therefore, while writing to this register, the SWR bit must be cleared to “0”. Otherwise a Software reset will be performed.

Here some examples for watchdog trigger commands:

- *Watchdog triggering in SBC Normal Mode, watchdog Timeout Mode and watchdog period of 16 ms:*
  - *SPI command: 0000**1001**0XXXXXXX$_b$*
- *Watchdog triggering in SBC Normal Mode, watchdog Window Mode and watchdog period of 8 ms:*
  - *SPI command: 0000**0000**0XXXXXXX$_b$*

- *Watchdog triggering in SBC Standby Mode, watchdog Timeout Mode and watchdog period of 256 ms:*
    - *SPI command: 0000**0101**0XXXXXXX$_b$*
- *Watchdog triggering in SBC Standby Mode, if watchdog shall be disabled afterwards:*
    - *SPI command: 0000**1XXX**0XXXXXXX$_b$*

It must be recognized that the watchdog mode does not only change by writing the WMC bit, but also when a SBC Mode change is performed. Table 11 shows the relation between WMC bit and the Mode Control (MC) bit in the Mode Control Register. Moreover, it must be considered, that a change of WMC in Normal Mode will immediately cause a system reset due to system failsafe reasons. Hence all watchdog mode configurations must be done in Standby Mode.

Example 1: The SBC is in Standby Mode with watchdog Timeout Mode (WMC = 0) configured (NWP = 16 ms) and then enters Normal Mode (MC = 2 or 3). Simultaneously with entering Normal Mode the watchdog is configured to work in Window Mode. Nevertheless, the watchdog trigger remains the same as before entering Normal Mode:

*SPI command: 0000**0001**0XXXXXXX$_b$*

The same mechanism is also valid for the opposite direction.

Example 2: The SBC is in Standby Mode with watchdog Off Mode (WMC = 1) configured and then enters Normal Mode (MC = 2 or 3). Simultaneously with entering Normal Mode the watchdog is configured to work in Timeout Mode. If the application requires the watchdog to work in Window Mode instead (e.g. NWP = 32 ms), the WDSR must be configured before entering the Normal Mode (WMC must be changed from 1 to 0). For the new watchdog configuration as well as for the triggering in Normal Mode following command has to be used:

*SPI command: 0000**0010**0XXXXXXX$_b$.*

Furthermore, it must be considered that changing the WMC bit in Standby Mode is only possible if no interrupt is pending. Any attempt to change the WMC bit in Standby Mode when an interrupt is pending will be ignored.

When the watchdog shall be used in Timeout mode during normal operation, i.e. while the SBC is in Normal mode, the WMC bit needs to be set. Setting WMC is only possible while the SBC is in Standby mode. However, in Standby mode WMC=1 means watchdog off, and therefore any new interrupt would clear WMC again. If interrupts have not been disabled before, an interrupt could occur right between setting WMC and switching to Normal mode. Therefore the status of WMC would have to be checked right after entering Normal mode. The procedure for setting WMC would have to be repeated until WMC=1 can be confirmed in Normal mode. Accordingly, interrupts should be disabled until WMC has been set and the SBC is Normal mode.

2.9.4 Watchdog triggering

In Timeout Mode the watchdog can be reset at any time within the watchdog period by a watchdog trigger. If the watchdog overflows due to a missing trigger event, the cyclic interrupt (CI) bit is set. If a CI is already pending, a reset is performed.

In Window Mode the watchdog can only be reset in the second half of the watchdog period (“open window”) by a watchdog trigger. If the watchdog is triggered in the first half of the watchdog period (“closed window”) or overflows, a system reset is immediately performed.

To select the correct trigger moment the tolerances of the watchdog timer must be taken into account. The software has to send the trigger signal after the latest possible window opening and before the earliest end of a period. For additional information see chapter 3.2.2.

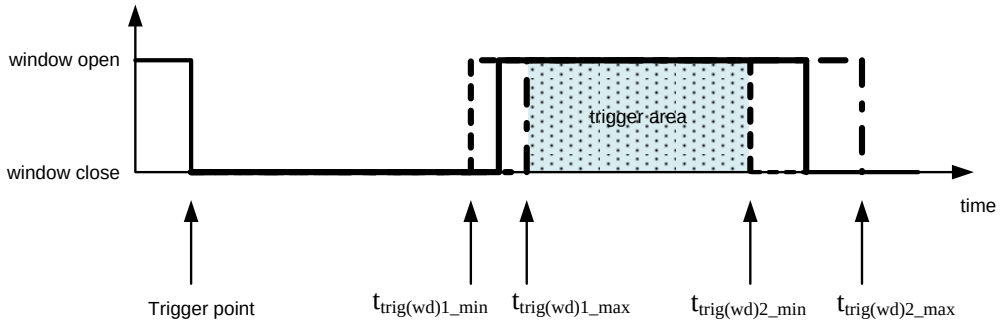


Fig 43. Allowed trigger area, when watchdog is in Window Mode

As shown in Fig 43, the trigger moment has to be located between the maximum value of $t_{trig(wd)1}$ (earliest watchdog trigger point) and the minimum value of $t_{trig(wd)2}$ (latest watchdog trigger point). The related values can be found in the data sheet [1]. In case the tolerances of the microcontroller's clock generator cannot be neglected, the following equations should be used to define the correct trigger moment:

$$t_{trigger\_\min} > \frac{t_{trig(wd)1\_max}}{1-F}, \qquad t_{trigger\_\max} < \frac{t_{trig(wd)2\_\min}}{1+F}.$$

F here is the magnitude of relative deviation of the microcontroller’s clock frequency. It is calculated by:

$$F = \left| \frac{f - f_0}{f_0} \right|,$$

where f is the actual frequency and f_0 is the nominal frequency. $t_{trigger\_min}$ and $t_{trigger\_max}$ limit the area where the software has to choose the trigger point supposing the nominal frequency of the clock generator.

A watchdog trigger is performed with every valid write access to the Watchdog and Status Register. After a successful SPI transfer the watchdog is reset. The transmitted value for the Nominal Watchdog Period NWP (see chapter 2.9.3) defines the duration of the next watchdog cycle. This is illustrated in Fig 44.

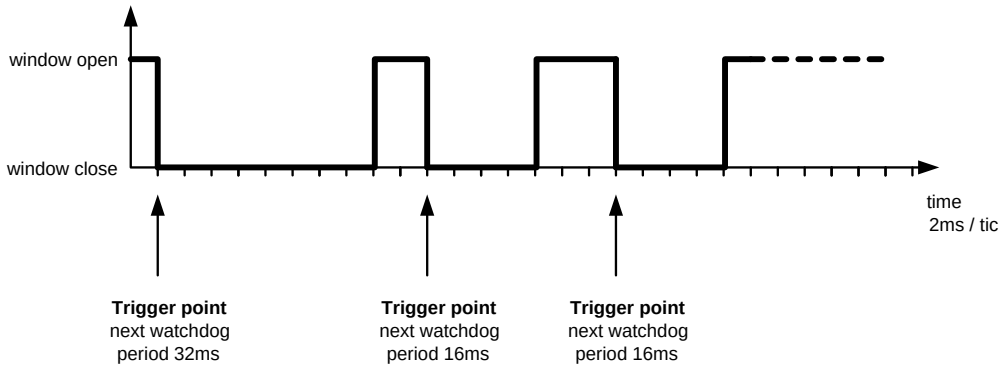


Fig 44. Watchdog triggering with different periods

2.9.5 Characteristics for UJA107xA types with hardware deactivated watchdog

There are UJA107xA types without watchdog available (see Table 2). Additionally for all types containing a watchdog the complete watchdog functionality can be disabled by connecting the WDOFF pin to V1 (see Fig 42). In addition to the deactivated watchdog functionality, all devices without watchdog or with disabled watchdog (WDOFF connected to V1) are characterized by the following features:

- The WOS bit is always set and read back as “1”.
- The NWP and WMC bits can be written and read back, but these bits have no effect on SBC behavior. That means for instance that a change of the WMC bits in Normal Mode will not trigger a reset.
- The cyclic interrupt feature is not available, as it is based on the watchdog timer that is deactivated.
- The V1 and RSTN clamped low detection is not supported, as it uses the watchdog timer that is disabled.

2.10 Special applications of UJA107xA

2.10.1 Transmission of slow signals via LIN Interface

As LIN 2.1 does specify a minimum transmission rate of 1 kbit/s, the UJA107xA LIN Interface is normally not intended to transmit very slow signals. Transmission of long LOW sequences is limited by the UJA107xA TXDL dominant timeout that cannot be disabled. In general applications the TXDL dominant time-out timer circuit prevents the bus lines being driven to a permanent dominant state (blocking all network communications), if TXDL is forced permanently LOW by a hardware and/or software application failure. Therefore for slow transmission rates the TXDL dominant time-out function has to be overcome.

As the LIN TXD dominant timeout is reset as soon as there is a positive edge on the TXDL pin, the solution is to set the TXDL input signal about every 10 ms to 20 ms for a short moment on HIGH level (about 1 µs to 5 µs). The LIN bus is too slow to follow the signal but it will reset the TXD dominant timer (see Fig 45).

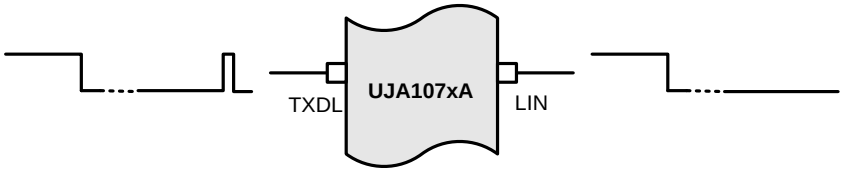
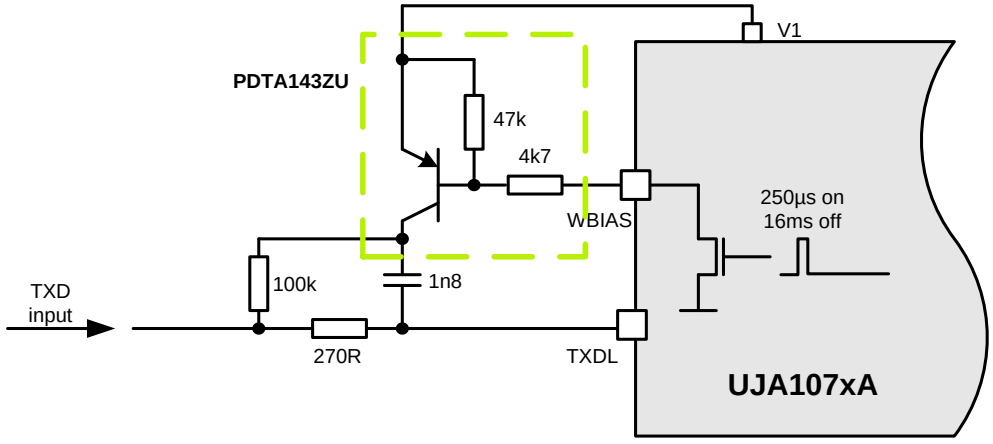


Fig 45. Support of low LIN transmission rates

This idea can be implemented by:

1. Providing an input signal by µC software (e.g. interrupt) or
2. Using the hardware workaround shown Fig 46 Here the SBC WBIAS output is programmed to provide a signal with 16 ms period. This signal triggers a transistor that pulls the TXDL input for a short moment on HIGH level via a capacitor.



UJA107xA Register Settings

Mode Control Register:

(1) WBC = 0

(2) LSC = 1

Interrupt Control Register:

(1) WSE1 = 1

Fig 46. Hardware circuitry to support transmission of slow signals via LIN Interface

With use of either workaround 1 or 2 even transmitting and receiving with low transmission rates is possible.

2.11 Pin FMEA

The pinning of the UJA107xA ensures that shorts between adjacent pins do not result in a system dead lock. Following classes of severity are distinguished:

Table 12. Classification of severity of effects

Severity Class	Effects
A	- Damage to device - Serial communication on CAN and/or LIN may be affected globally
B	- No damage to device - Serial communication in the overall system not possible (global problem)
C	- No damage to device - Bus communication of other nodes in the system possible - Corrupted node not able to communicate (local problem) - Application might shut-down
D	- No damage to device - Bus communication in the overall system possible - Reduced functionality of application
-	- Not affected at all

Table 13. Pin FMEA

Pin Name	Failure	Remark	Severity Class
TXDL2 (only applicable for UJA1078A)	Shorted to neighbor (RXDL2)	RXDL2 follows TXDL2; transmission on LIN2 possible. Consider functionality of connected MCU pin.	D
	Shorted to GND	LIN2 stays recessive all time due to TXDL dominant timer protection; No LIN2 communication. Consider functionality of connected MCU pin.	C
	Shorted to V1	LIN2 stays recessive all time; No LIN2 communication. Consider functionality of connected MCU pin.	C
	Open Circuit	LIN2 stays recessive all time; No LIN2 communication	C
RXDL2	Shorted to neighbor (TXDL1)	RXDL2 follows TXDL1; transmission on LIN1 and LIN2	C/D

Pin Name	Failure	Remark	Severity Class
(only applicable for UJA1078A)		possible	
	Shorted to GND	LIN2 transmission possible; LIN2 reception not possible	D
	Shorted to V1	LIN2 transmission possible; LIN2 reception not possible	D
	Open circuit	LIN2 transmission possible; LIN2 reception not possible	D
TXDL1 (not applicable for UJA1076A)	Shorted to neighbor (V1)	LIN1 stays recessive all time; No LIN1 communication. Consider functionality of connected MCU pin.	C
	Shorted to GND	LIN1 stays recessive all time due to TXDL dominant timer protection; No LIN1 communication. Consider functionality of connected MCU pin.	C
	Open circuit	LIN1 stays recessive all time; No LIN1 communication	C
V1	Shorted to neighbor (RXDL1)	LIN1 stays recessive all time; No LIN1 communication	C
	Shorted to GND	V1 off; permanent system reset; LIMP home activated	C
	Open circuit	Microcontroller un-powered	C
RXDL1 (not applicable for UJA1076A)	Shorted to neighbor (RSTN)	LIN message from other nodes would cause repeated reset events	C
	Shorted to V1	LIN1 transmission possible; LIN1 reception not possible	D
	Shorted to GND	LIN1 transmission possible; LIN1 reception not possible	D
	Open circuit	LIN1 transmission possible; LIN1 reception not possible	D
RSTN	Shorted to neighbor (INTN)	System interrupt causes permanent system reset	C
	Shorted to V1	No system reset possible	C
	Shorted to GND	Permanent system reset; LIMP home activated	C
	Open circuit	No system reset via SBC possible	C
INTN	Shorted to neighbor (EN)	No interrupts supported; unexpected (permanent) interrupts	D

Pin Name	Failure	Remark	Severity Class
	Shorted to V1	No interrupts supported	D
	Shorted to GND	Permanent interrupt	D
	Open circuit	No interrupts supported	D
EN	Shorted to neighbor (SDI)	No communication towards SBC	C
	Shorted to V1	No protection of critical application hardware; hardware is always active	D
	Shorted to GND	Safety critical application hardware might be disabled all time	D
	Open circuit	Safety critical application hardware might be disabled all time	D
SDI	Shorted to neighbor (SDO)	No communication towards SBC. Consider functionality of connected MCU pins.	C
	Shorted to V1	No communication towards SBC. Consider functionality of connected MCU pin.	C
	Shorted to GND	No communication towards SBC. Consider functionality of connected MCU pin.	C
	Open circuit	No communication towards SBC	C
SDO	Shorted to neighbor (SCK)	No communication towards SBC. Consider functionality of connected MCU pin.	C
	Shorted to V1	Cannot read from SBC; behavior depends on software	C / D
	Shorted to GND	Cannot read from SBC; behavior depends on software	C / D
	Open circuit	Cannot read from SBC; behavior depends on software	C / D
SCK	Shorted to neighbor (SCS)	No communication towards SBC. Consider functionality of connected MCU pin.	C
	Shorted to V1	No communication towards SBC. Consider functionality of connected MCU pin.	C

Pin Name	Failure	Remark	Severity Class
	Shorted to GND	No communication towards SBC. Consider functionality of connected MCU pin.	C
	Open circuit	No communication towards SBC	C
SCSN	Shorted to neighbor (TXDC)	Interruptions in communication towards SBC possible; sporadic CAN error frames possible. Consider functionality of connected MCU pins.	C
	Shorted to V1	No communication towards SBC. Consider functionality of connected MCU pin.	C
	Shorted to GND	No communication towards SBC. Consider functionality of connected MCU pin.	C
	Open circuit	No communication towards SBC	C
TXDC (not applicable for UJA1079A)	Shorted to neighbor (RXDC)	RXDC follows TXDC; CAN transmission possible. Consider functionality of connected MCU pin.	C
	Shorted to V1	No transmission of CAN messages possible; node runs bus off; no effect on the communication of the other nodes; reception still possible. Consider functionality of connected MCU pin.	D
	Shorted to GND	SBC detects this short and disables the CAN transmitter; no effect on other nodes. Consider functionality of connected MCU pin.	C
	Open circuit	No transmission of CAN frames possible; CAN controller runs bus off; no effect on other nodes communication; reception still possible	D
RXDC (not applicable for UJA1079A)	Shorted to neighbor (TEST1)	No effect. If TEST1 is connected to GND, see “Shorted to GND”.	-
	Shorted to V1	Short-term disturbance of CAN communication until CAN error counter reaches limit and CAN protocol engine stops CAN transmission; bus is off; other nodes not affected	C
	Shorted to GND	Short-term disturbance of CAN communication until CAN error counter reaches limit and CAN protocol engine stops CAN transmission; bus is off; other nodes	C

Pin Name	Failure	Remark	Severity Class
		not affected	
	Open circuit	Short-term disturbance of CAN communication until CAN error counter reaches limit and CAN protocol engine stops CAN transmission; bus is off, other nodes not affected	C
TEST1	Shorted to neighbor (WDOFF)	No test mode; Normal operation.	-
	Shorted to BAT	No test mode; Normal operation	-
	Shorted to GND	No test mode; Normal operation	-
	Open circuit	No test mode; Normal operation	-
WDOFF	Shorted to V1	Watchdog off; normal operation	-
	Shorted to GND	Watchdog on; normal operation	-
	Open circuit	Watchdog on/off (depends on previous state); normal operation	-
LIMP	Shorted to neighbor (Wake1)	Unexpected WAKE1 interrupts; unexpected activation of “Limp Home” hardware	D
	Shorted to BAT	“Limp Home” hardware is permanently off	D
	Shorted to GND	“Limp Home” hardware is permanently on	D
	Open circuit	No control of “Limp Home” hardware possible	D
WAKE1	Shorted to neighbor (WAKE2)	Unexpected wake-up interrupts; Wake source distinction between WAKE1 and WAKE2 not possible	D
	Shorted to BAT	No local wake-up possible	D
	Shorted to GND	No local wake-up possible	D
	Open circuit	No local wake-up possible	D
WAKE2	Shorted to neighbor (V2)	Unexpected wake-up interrupts on entering and leaving Normal Mode; V2 undervoltage in case of a closed switch to GND at WAKE2	D

Pin Name	Failure	Remark	Severity Class
	Shorted to BAT	No local wake-up possible	D
	Shorted to GND	No local wake-up possible	D
	Open circuit	No local wake-up possible	D
V2 (not applicable for UJA1079A)	Shorted to neighbor (CANH)	Single wire operation via CANL	-
	Shorted to BAT	SBC and other hardware is damaged	A
	Shorted to GND	SBC detects this short by V2 undervoltage; CAN transceiver offline; no CAN communication	C
	Open circuit	No buffering of V2; reception still possible; bus disturbances in transmission case	C
CANH (not applicable for UJA1079A)	Shorted to neighbor (CANL)	No CAN communication on complete bus possible (no differential signal possible)	B
	Shorted to BAT	Communication still possible (differential signal still available; CANL recessive level increased to BAT/V1 level)	-
	Shorted to GND	No CAN communication possible (no differential signal)	B
	Open circuit	No CAN communication possible (no differential signal)	C
CANL (not applicable for UJA1079A)	Shorted to neighbor (GND)	Communication still possible (differential signal still available; CANH recessive level lowered to GND level)	-
	Shorted to BAT	No CAN communication possible (no differential signal)	B
	Open circuit	No CAN communication possible (no differential signal)	C
GND	Shorted to neighbor (SPLIT, not applicable for UJA1079A)	Slight increase in system supply current; communication not affected; Single wire operation via CANH	D
	Shorted to BAT	Fundamental problem of ECU; no supply available; SBC not affected	
	Shorted to GND	Normal operation	-

Pin Name	Failure	Remark	Severity Class
	Open circuit	Fundamental problem of ECU; no supply available; SBC and components on ECU might become damaged	(A)
SPLIT (not applicable for UJA1079A)	Shorted to neighbor (LIN)	LIN recessive voltage slightly decreased; some bias current will flow but communication not affected	D
	Shorted to BAT	Slight increase in system supply current; communication not affected; single wire operation via CANL	D
	Shorted to GND	Slight increase in system supply current; communication not affected; single wire operation via CANH	D
	Open circuit	DC stabilization not active; lower DC stabilization effect in system	D
LIN1 (not applicable for UJA1076A)	Shorted to neighbor (DLIN)	LIN1 communication might be affected due to low-ohmic termination to BAT; low-power modes still possible	C
	Shorted to BAT	LIN1 communication down; low-power modes still possible; other nodes cannot communicate any more	B
	Shorted to GND	LIN1 communication down; low-power modes still possible; other nodes cannot communicate any more	B
	Open circuit	LIN1 communication down; low-power modes still possible	C
DLIN	Shorted to neighbor (LIN2; only applicable for UJA1078A)	LIN2 communication might be affected due to low-ohmic termination to BAT; LIN1 communication not affected; low-power modes still possible	D
	Shorted to BAT	BAT is default level of DLIN; loss of protection feature against unwanted currents on LIN	D
	Shorted to GND	No communication possible due to loss of termination	C
	Open circuit	No LIN bus communication possible due to loss of termination (only applicable for master applications)	C
LIN2 (only applicable for UJA1078A)	Shorted to neighbor (WBIAS)	Unexpected dominant signal on LIN2; unexpected WBIAS signals; LIN2 communication possible	D
	Shorted to BAT	LIN2 communication down; low-power modes still	B

Pin Name	Failure	Remark	Severity Class
		possible; other nodes cannot communicate any more	
	Shorted to GND	LIN2 communication down; low-power modes still possible; other nodes cannot communicate any more	B
	Open circuit	LIN2 communication down; low-power modes still possible	C
WBIAS	Shorted to neighbor (VEXCC)	If external PNP is used, biasing of switches is not possible as WBIAS is clamped to V1 level	D
	Shorted to BAT	WBIAS cannot be activated	D
	Shorted to GND	WBIAS permanently activated	D
	Open circuit	Connected switches cannot be biased via WBIAS	D
VEXCC	Shorted to neighbor (TEST2)	No effect. If TEST2 is connected directly to GND, see “Shorted to GND”.	-
	Shorted to V1	Short-circuit protection for PNP is lost (PNP current limiting activation threshold is never reached); PNP might be damaged	D*
	Shorted to GND	PNP is not activated; additional load at V1 due to short of VEXCC to GND causes a V1 undervoltage	C*
	Open circuit	PNP is never activated; Total current must be delivered out of V1; might cause an overtemperature shutdown	C*
TEST2	Shorted to neighbor (VEXCTRL)	No test mode; Normal operation. If TEST2 is connected directly to GND, see “VEXCTRL – Shorted to GND”.	-
	Shorted to BAT	No test mode; Normal operation	-
	Shorted to GND	No test mode; Normal operation	-
	Open circuit	No test mode; Normal operation	-
VEXCTRL	Shorted to neighbor (BAT)	PNP is never activated; Total current must be delivered out of V1; might cause an overtemperature shutdown	C*
	Shorted to GND	PNP might be damaged due to high basis current; V1	A*

Pin Name	Failure	Remark	Severity Class
		level temporarily raised above specification (up to BAT), increased current consumption; battery voltage drop; SBC not supplied	
	Open circuit	PNP is never activated; Total current must be delivered out of V1; might cause an overtemperature shutdown	C*
BAT	Shorted to BAT	Normal operating condition	-
	Shorted to GND	Short to GND is a fundamental ECU problem; SBC not affected. Polarity protection diode may get damaged. Fuse may blow.	(A)
	Open circuit	SBC not supplied	C

*only, if V1 is operated with an external PNP transistor. Without this PNP and without the resistor between VEXCC and V1, the effect is "-" (not affected at all)

2.12 Printed Circuit Board

The SBC UJA107xA is delivered in a HTSSOP32 package with an exposed die pad. In order to prevent overheating of the SBC it is strongly recommended to solder the heatsink to the PCB. This way, the thermal resistance R_{th}(j-a) can be improved significantly by approximately 20 K/W.

Furthermore, additional PCB layout options improve the thermal resistance:

- Number of PCB layers (best case: 4 layers)
- Cu thickness (35 µm / 70 µm; the thicker the better)
- Heatsink area on the PCB available for the SBC (the bigger the better)
- Number of vias (the more the better)

Moreover, an increase of the heatsink area at the top layer is more efficient than an extension of the heatsink on the bottom layers. Hence, for a good thermal performance of the PCB it is recommended to exploit as much area on the top layer as heatsink as possible and use the other layers for further optimization.

On demand UJA107xA thermal simulation models for computational fluid dynamics software can be provided.

2.13 Comparison of UJA107xA with UJA106x

The UJA107xA family is basically derived from the predecessor UJA106x SBC family. While the UJA106x is designed to offer a SBC with special system failsafe features, the UJA107xA family is reduced to the minimum ECU needs. The UJA107xA is technically improved and allows an easy replacement of the UJA106x family and vice versa. Table 14 shows a comparison of both SBC families regarding their features.

Table 14. Comparison of UJA107xA with UJA106x

Feature	UJA107xA	UJA106x
SBC Modes	Normal, Standby, Sleep	Normal, Standby, Sleep
Watchdog Modes	Window / Timeout / Off	Window / Timeout
Watchdog periods	8	40
Cyclic wake-up	Yes, cyclic wake-up from standby	Yes, cyclic wake-up from sleep and standby
Wake-up ports	2, one or both edges	1, falling edge
Limp Home	Yes	Yes
Supply voltage	14V	14V / 42V
Minimum battery voltage	4.5V	5.5V
Standby current Sleep current	~ 75 µA (standby) ~ 55 µA (sleep)	< 300 µA (standby) < 50 µA (sleep)
Reset thresholds	70% / 90%	70% / 80% / 90%
SPI	3 MBit/s, 16 bit	1 MBit/s, 16 bit
Battery sense (SENSE pin)	No	Yes
External DCDC control (SYSINH)	No	Yes
Battery switch (V3)	No	Yes
Partial networking	No	Yes
Extended failsafe modes / behavior	basic failsafe	Yes
IEC 61000-4-2 (150pF / 330 Ohm) Human body model (100pF/1.5 kOhm)	6kV 8KV	4kV 8KV
Voltage regulator	250mA	120mA
Heat distribution with external pnp transistor	Yes	No
Bus failure diagnosis on CAN and LIN	No	Yes
Autonomous fail-safe termination on CAN and LIN	No	Yes

Feature	UJA107xA	UJA106x
Ground shift diagnosis	No	Yes
Detection of cyclic failures with General Purpose Registers	No	Yes
Detailed device identification	No	Yes
Flash update	Normal mode w/ timeout watchdog	Flash mode
Overtemperature protection	Overtemperature shutdown	Overtemperature warning

Even the package and the pinning of the UJA107xA and UJA106x are similar to allow an easy exchange of both SBC families without additional changes on the PCB layout. Fig 47 shows the pinning of the both HSCAN-LIN SBC derivatives. The differences result from:

- The possibility to use an external PNP transistor with the UJA107xA for better heat distribution (see 2.4.2).
- The WBIAS pin that controls the bias transistor of the switches (see 2.7.4).
- The introduction of the WDOFF pin for disabling the Watchdog completely for easy software development (see 2.9.2).
- The replacement of the INH/LIMP pin by only a LIMP pin (active-LOW, open-drain output pin).

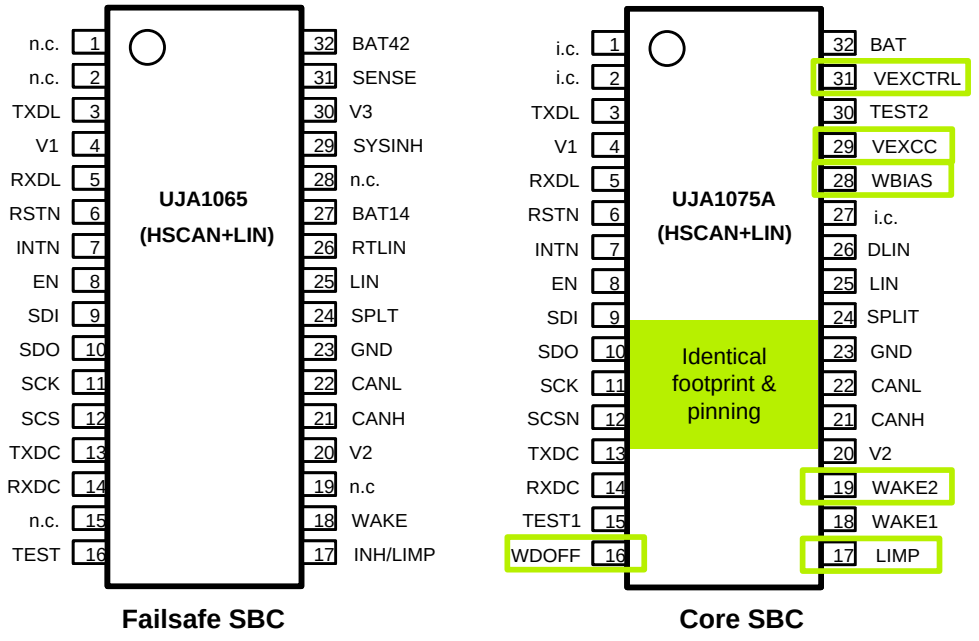


Fig 47. UJA1075A - UJA1065 pinning comparison

In general the application circuit for UJA107xA and UJA106x is the same. Only a few adaptations are necessary to use alternatively an UJA106x or an UJA107xA with all features within the same PCB layout.

Fig 48 and Fig 49 illustrate the common circuit and the recommended adaptations in population for an UJA1065 and an UJA1075A.

As it is recommended to connect the TEST pin of the UJA106x via a pull-down resistor (e.g. 1kΩ) to ground in the application anyway, there is no difference to the common UJA107xA circuit where the watchdog is activated per default. Furthermore, also the external LIMP Home circuitry does not deviate. In case the wake-up switches shall be sampled cyclically, with the UJA106x the cyclic sensing feature is realized by use of the V3 output. For the UJA107xA the WBIAS output in combination with an external bias transistor is necessary to realize cyclic sampling of wake-up switches (see Fig 48 and Fig 49). Apart from that the circuitry only needs further adaptations in case an external PNP transistor shall be used to improve the heat distribution. Additionally, it must be considered that the UJA107xA TEST pins (TEST1 and TEST2) should be connected to ground in the application (either directly or via a resistor < 11 kΩ).

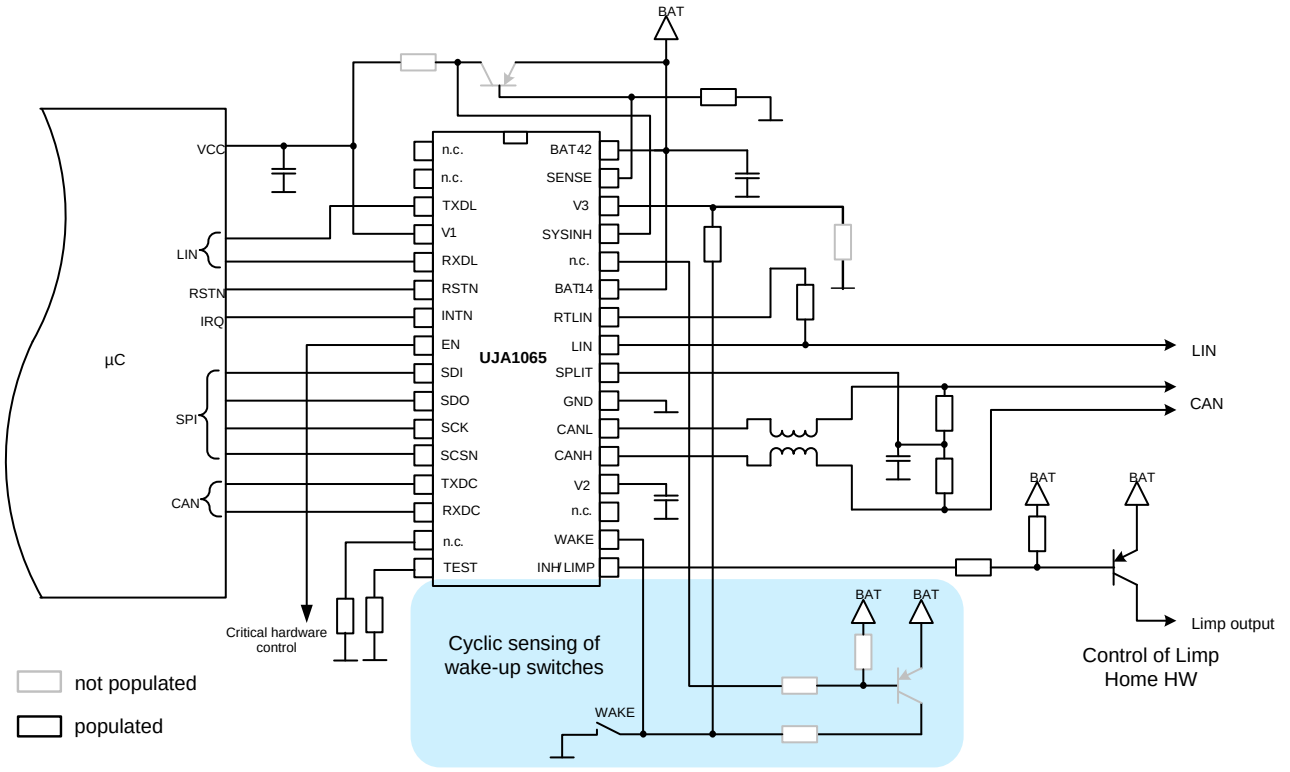


Fig 48. Common circuit for UJA1075A and UJA1065 - Population for Failsafe SBC UJA1065

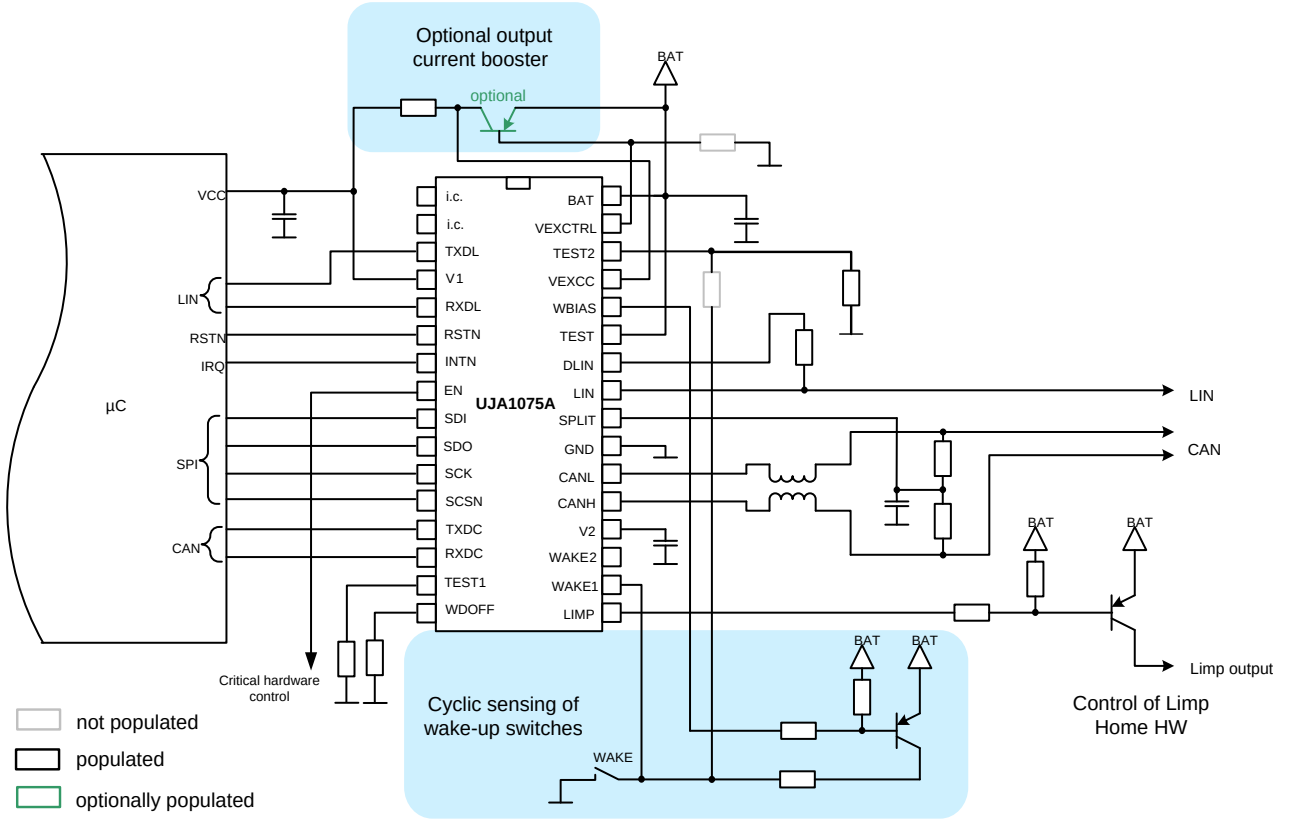


Fig 49. Common circuit for UJA1075A and UJA1065 - Population for Core SBC UJA1075A

3. Software Flow

This chapter introduces the software perspective of the UJA107xA family. It discusses the different operations which are used in automotive applications. Fig 50 illustrates the different operations between “Power-on” and “Power-off”. Moreover, the figure below provides a quick overview about:

- Different kind of operations
- Order of the operations
- Subchapter where the operation is discussed in more details.

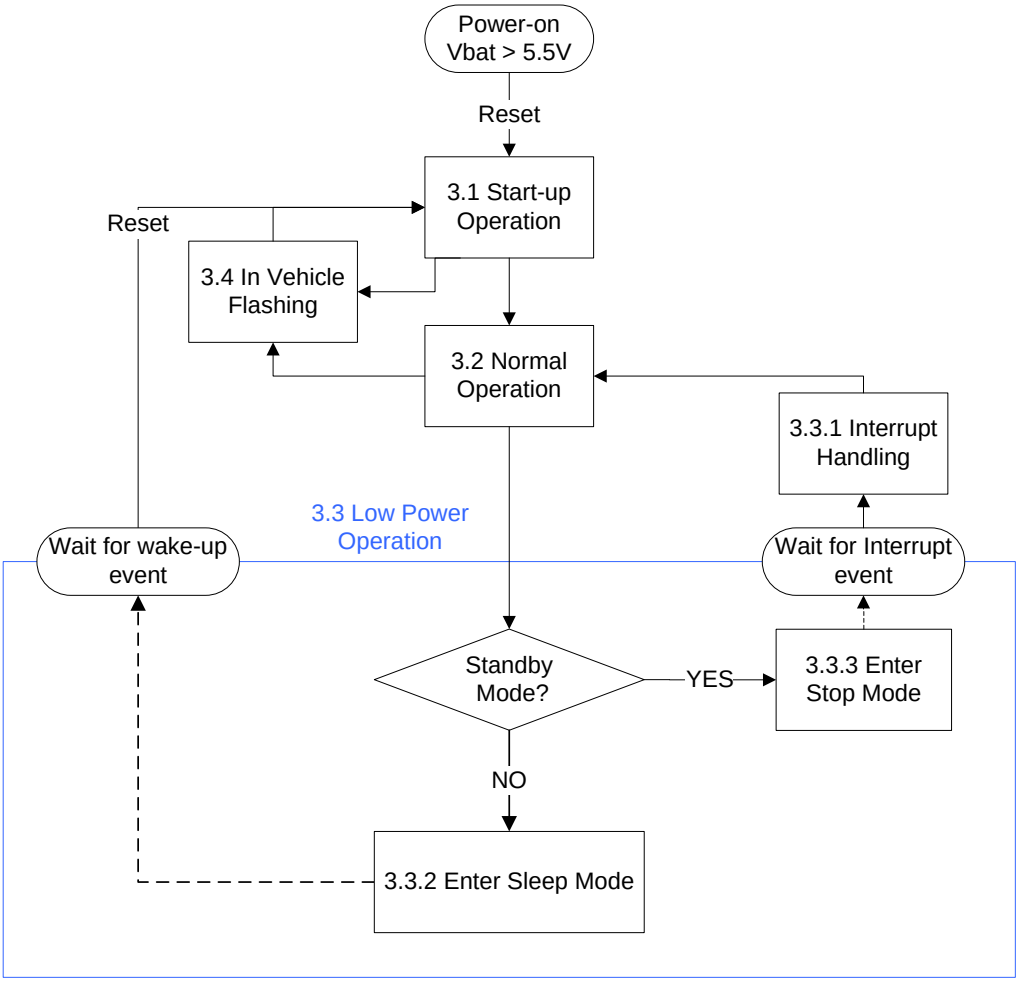


Fig 50. Abstract view on software flow

Example code of all different operations can be found in the appendix chapter 0.

The figure below shows the simplified state diagram of the UJA107xA family. This state diagram will be used in the following subchapters to establish the link between the different software operations and the related operating modes of the UJA107xA.

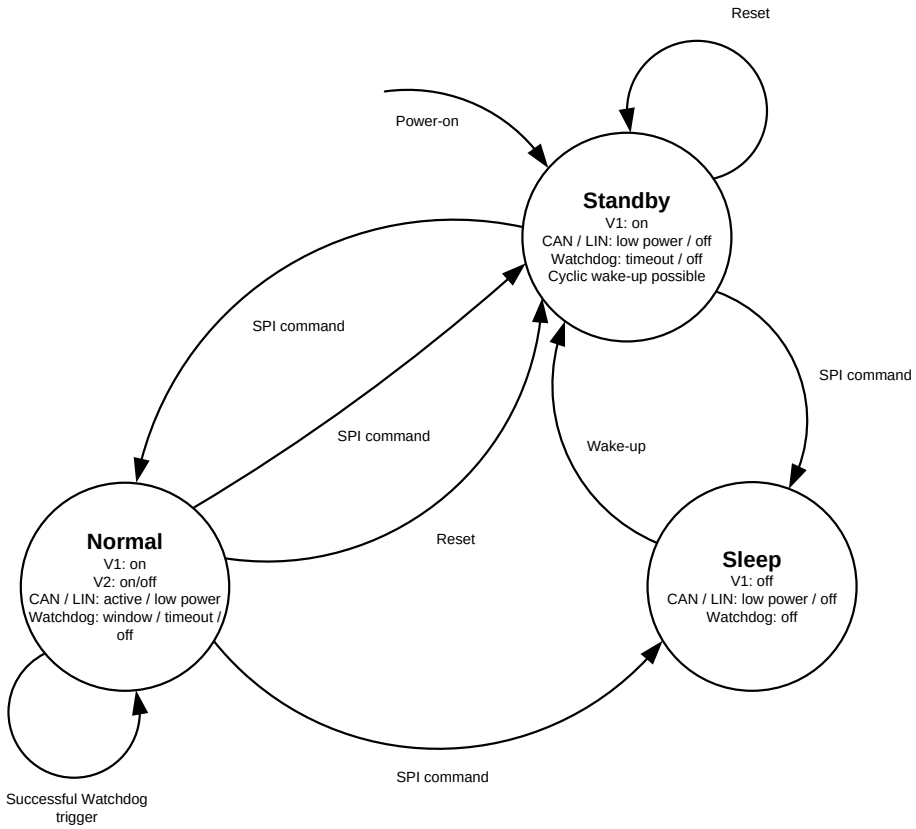


Fig 51. Simplified mode diagram of UJA107xA Family

3.1 Startup Operation

This section introduces the software operations which are related to the startup of the application. The Startup Operation is always executed after the release of RSTN and takes place in Standby Mode of the UJA107xA (see picture below). Moreover, the figure below shows the different hardware events that trigger the execution of the Startup Operation. All kinds of resets (e.g. external reset, watchdog failure, V1 undervoltage etc.) trigger the execution of the startup operation. Furthermore, the “Power-on” and “Wake-up” hardware events trigger the Startup Operation as well. At the end of the Startup Operation a transition to Normal Mode is performed via the related SPI command.

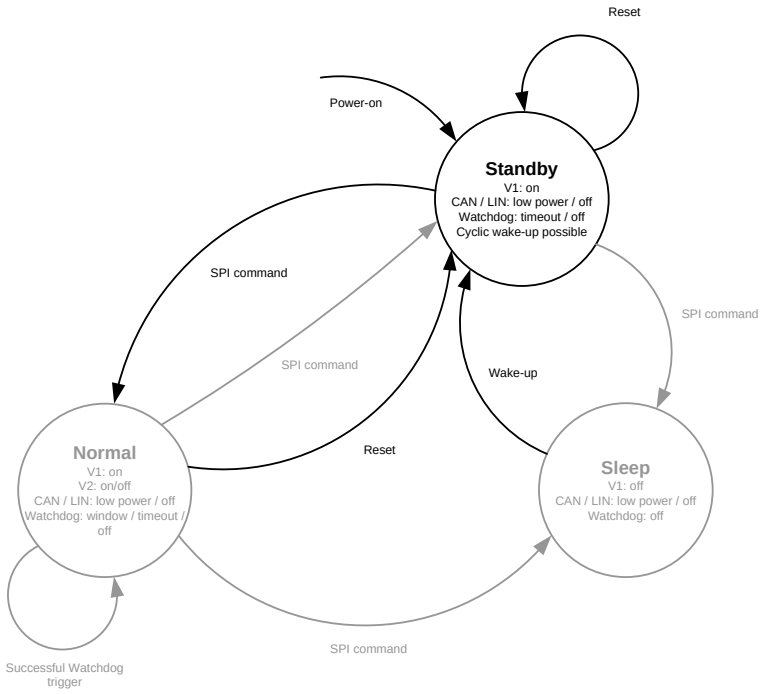


Fig 52. Simplified mode diagram with focus on startup operation

The Startup Operation typically consists of the following parts:

- In-vehicle Flashing of ECUs (if implemented)
- Microcontroller initialization
- Watchdog configuration
- Reset and Wake-up source detection
- Limp Home handling

- Transition to Normal Operation

Fig 53 shows the complete flow of the Startup Operation with its different parts. Moreover, it guides to the related subchapter for a detailed explanation.

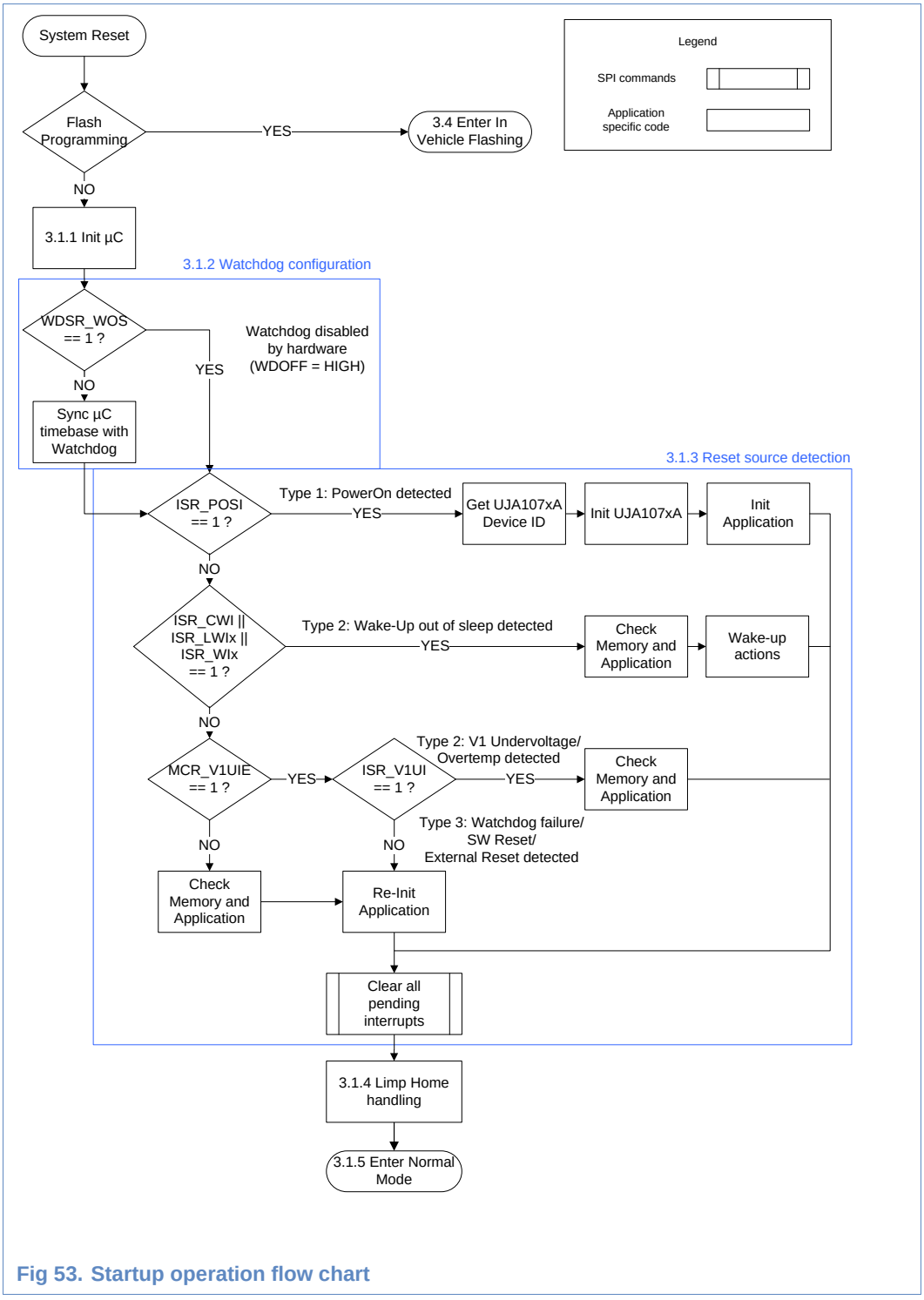


Fig 53. Startup operation flow chart

3.1.1 Microcontroller initialization

The microcontroller initialization is always the first part of the startup routine. It is a microcontroller specific routine that configures the microcontroller and its periphery. Therefore, it has to be ensured that after the initialization the SPI periphery of the microcontroller is working properly. Otherwise no communication with the UJA107xA can take place.

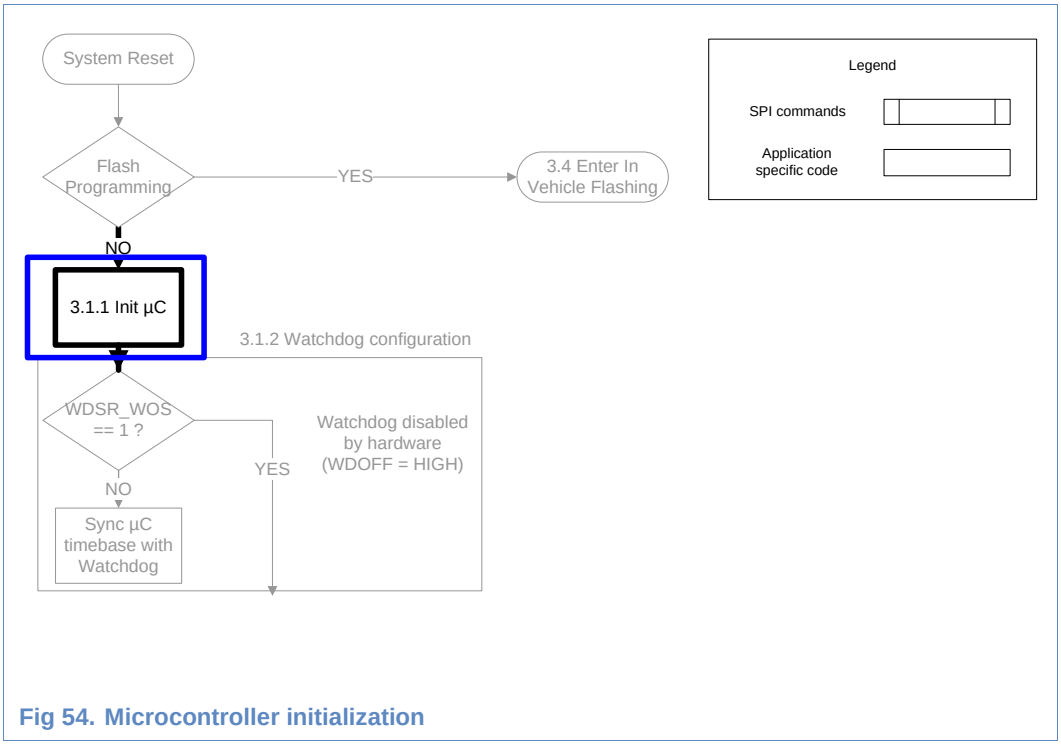


Fig 54. Microcontroller initialization

3.1.2 Watchdog configuration

The second part of the Startup Operation is the watchdog configuration. This part must be executed within 256ms after the reset is released. Otherwise a watchdog overflow reset can be triggered and the Startup Operation restarts.

The UJA107xA family provides possibilities to find out if the watchdog is running or disabled. Therefore, a read access to the Watchdog and Status Register (WDSR) must be performed (SPI command = 0x1000). The 7th bit of the register is called Watchdog Off Status (WOS) and indicates if the watchdog is running or not. If WDSR_WOS = 1, the watchdog is disabled by hardware and no watchdog related resets are possible. Therefore, no watchdog triggers are required.

If WDSR_WOS = 0, the microcontroller time base and the UJA107xA watchdog timer have to be synchronized in order to allow a proper watchdog service in software, especially for the Window Mode. After the synchronization the watchdog can be triggered with the required period. The previous described flow can also be seen in the figure below.

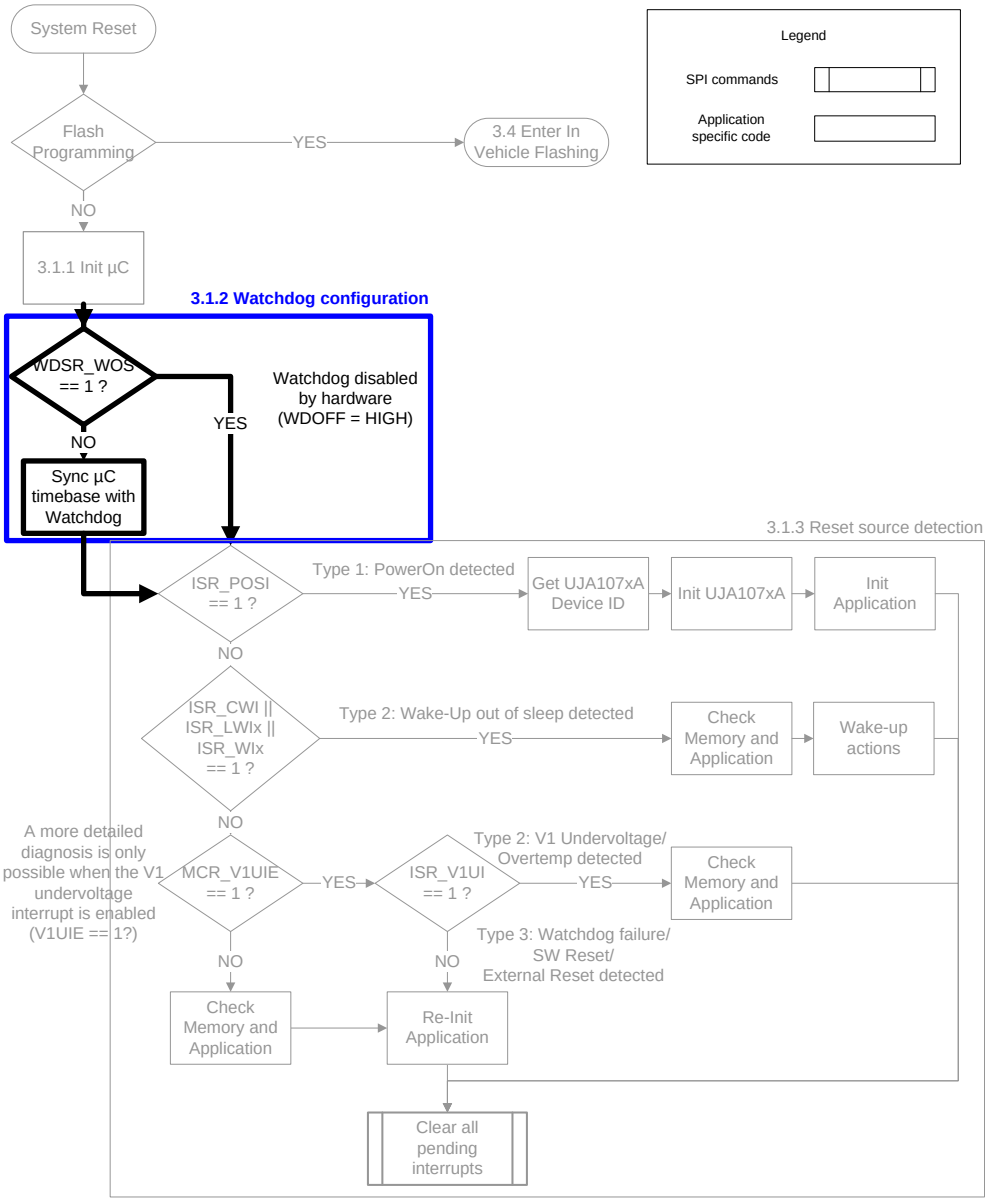


Fig 55. Watchdog configuration

Fig 56 illustrates why synchronization between the UJA107xA watchdog timer and the microcontroller time base is required. Background is that with a cold start of the system both time domains are completely unsynchronized since timers in the SBC as well as timers in the microcontroller start at different moments in time. The synchronization mechanism depends on the general software architecture of the application. In general two strategies exist, how to synchronize both timers or rather time domains.

1. Restart/Start the microcontroller time base short before a watchdog trigger (Synchronize microcontroller timer)
2. Trigger the watchdog with respect to the microcontroller time base (Synchronize Watchdog timer)

Strategy 1 is the most common one in applications without Operating System (OS). Here a dedicated time base within the microcontroller can be adapted to the watchdog time base in the SBC. The situation is different in applications containing a time triggered OS. The OS time base cannot be reset during operation, because this would corrupt the entire software schedule. Therefore strategy 2 is required synchronizing the SBC time base towards the already running operating system time base.

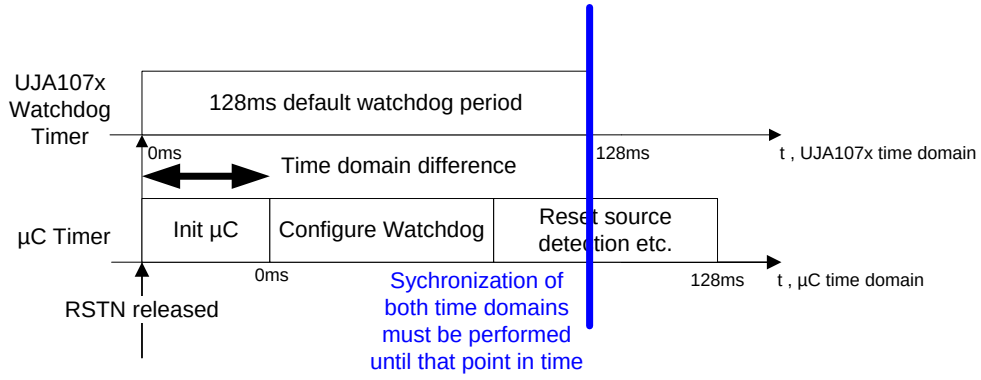


Fig 56. Microcontroller and watchdog synchronization

In the picture above it is illustrated, that the SBC time domain starts with the release of the reset signal. After that, the microcontroller starts its software with a device specific and software specific delay (time domain difference). In order to give the application enough time to settle all peripherals and starting the Operating System, the SBC provides up to 128ms, until the watchdog needs his first service trigger. This first service has to come before the 128ms are elapsed (see blue line in the picture above). After the software has been started and the watchdog service is initialized, the first watchdog service takes place at the end of the “Configure Watchdog” phase. With this first trigger, the Watchdog is synchronized and starts to operate with the software defined nominal watchdog period.

For more information on how to trigger the UJA107xA watchdog refer to chapter 2.9.

3.1.3 Reset and Wake-up source detection

The next step in the Startup Operation is the reset and wake-up source detection. It depends on the application if reset and wake-up source detection is required. From UJA107xA perspective it is not necessarily needed but the UJA107xA provides the information to determine the reset and wake-up source.

If the reset and wake-up sources are not of interest the software flow can be seen in the figure below. It has to be ensured that all interrupt sources are cleared (to release INTN) and that an initialization of the UJA107xA is performed. If interrupts are still pending the INTN pin (connected to an external interrupt) cannot trigger any further software actions because it remains low. Moreover, memory checks (RAM and/or Flash) could be performed to ensure proper operation, and the dynamic memory content must be updated. To clear all pending interrupts within one access it is recommended to read the Interrupt Status Register (ISR) and clear only these bits which are set. For example the read access to ISR (SPI command 0x7000) reads back 0x7808 (V1UI and CWI pending). V1UI and CWI are cleared with the SPI command 0x6808. (Note: Reading Interrupt flags does not clear these flags. Clearing of flags is possible with a write access to the ISR register only.)

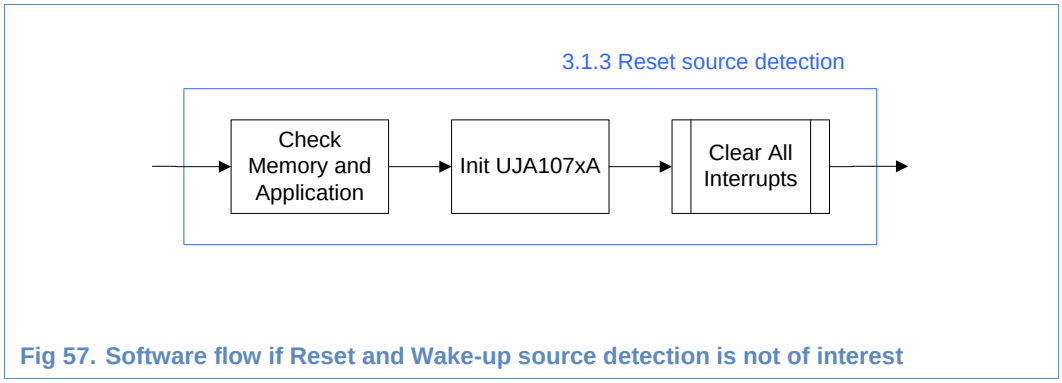


Fig 57. Software flow if Reset and Wake-up source detection is not of interest

If reset and wake-up sources are of interest, there are three different types of resets to be distinguished in the software flow:

- Type 1: Reset where the configuration of the UJA107xA and the microcontroller is lost
- Type 2: Reset where only the configuration of the microcontroller is lost and the RAM might be corrupted
- Type 3: Reset caused by a software failure

The software must be able to determine between the different types of resets, but it is not required to differentiate between resets of the same type because the related actions are the same.

In case of a *type 1 reset* an initialization of the UJA107xA is required. Moreover, memory checks (RAM and/or Flash) could be performed to ensure proper operation and the dynamic parameters must be updated. The only type 1 reset of the UJA107xA is the power-on reset.

In case of a *type 2 reset* memory checks (RAM and/or Flash) could be performed to ensure proper operation, and the dynamic parameters must be updated. Type 2 resets are all wake-ups out of Sleep Mode via CAN, LIN or WAKE, the V1 undervoltage reset and the overtemperature shutdown.

In case of a *type 3* reset the application needs to be checked and re-initialized. Type 3 resets are external resets, watchdog failures and software resets.

Fig 58 shows the detailed software flow for reset and wake-up source detection applicable for the UJA1078A.

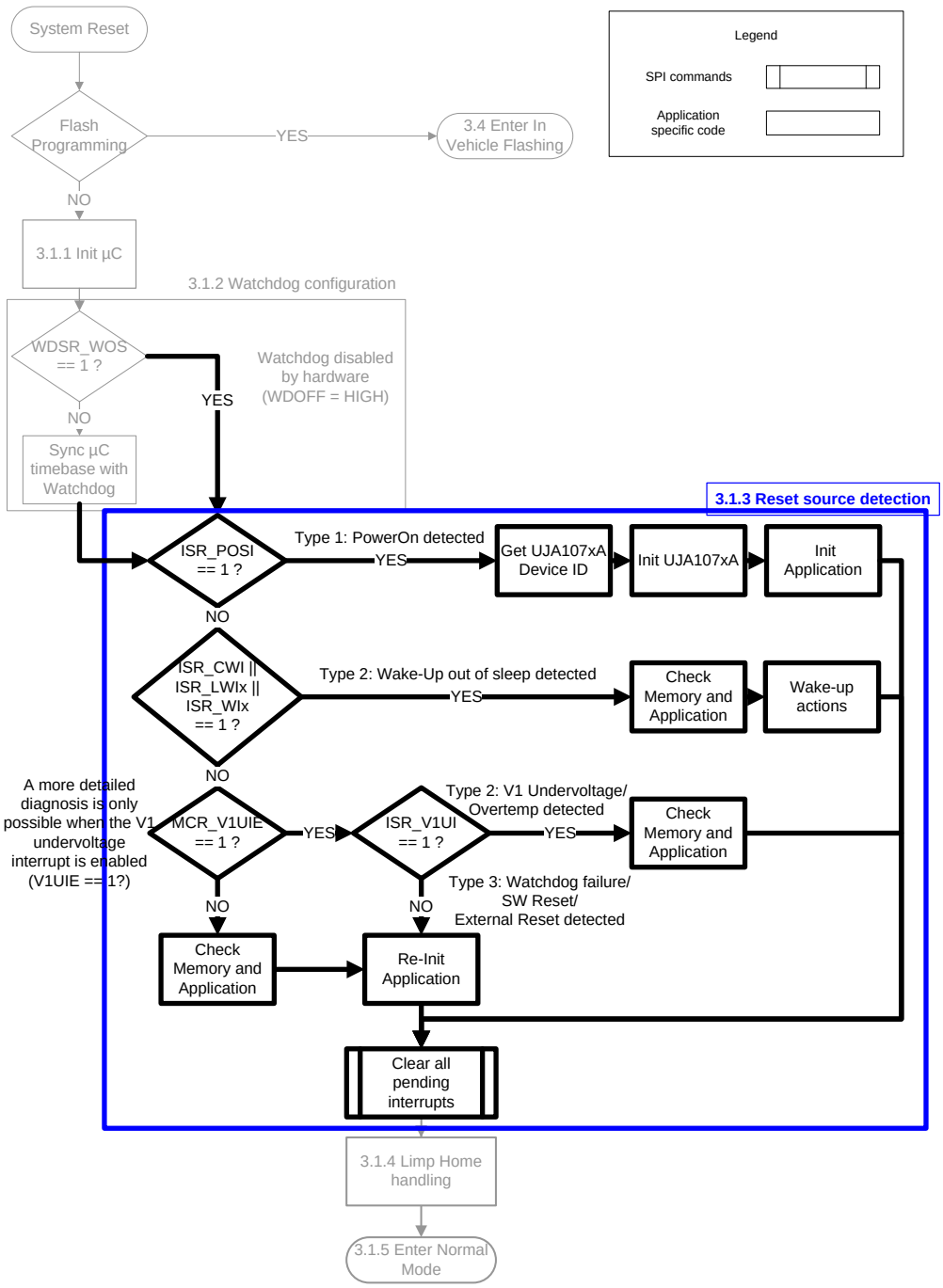


Fig 58. Reset source detection

Before all the checks can be performed, a read access to the Interrupt Status Register (ISR) and the Mode Control Register (MCR) must be performed. The SPI command for reading ISR is 0x7000 and for reading MCR is 0x3000. Please consider that the read access to ISR will not automatically clear the interrupt. For clearing the pending interrupts a write access to the ISR is required. (For more information see chapter 2.3.2.2.

The first step of the reset and wake-up source detection is the power-on detection. Power-on is indicated by the Power-On Status Interrupt (POSI). If this bit is equal to 1, a power-on is detected. In case of a power- on (type 1 reset) an initialization of the UJA107xA is required. The initialization is an application specific operation that defines the settings for e.g. LIN slope, WAKE sampling period, watchdog period, interrupt settings etc. At the end of the power-on related actions the POSI bit must actively be cleared. This is done by writing 0x6020 to ISR.

Furthermore, before initialization it is possible to identify which family member is used in the current application. It can be distinguished between UJA1078A, UJA1076A, UJA1075A and UJA1079A derivative of the family. It is not possible to identify if the used device is a 5V or a 3.3V device. The so-called device ID is provided implicitly by the Interrupt Control Register (ICR). The first step of getting the device ID is writing 0x4308 to ICR. This command simply tries to enable the wake-up receiver of the CAN and LIN Transceivers and based on their reaction, it can be checked, whether the UJA107xA derivative provides these interfaces or not. After writing the ICR it can be read back with the 0x5000 command. The read data of ICR now show the device ID (see Table 15). Moreover, it is optionally possible to write back the default configuration of the ICR in case transceiver wake-ups are not of interest directly after power-on.

Table 15. UJA107xA family devices

Derivative	Device ID
UJA1075A	0x5208
UJA1076A	0x5008
UJA1078A	0x5308
UJA1079A	0x5200

In case no power-on has been detected (POSI = 0), the next step in the software flow of Fig 58, is the wake-up source detection. Possible wake-up sources are CAN, LIN and wake pins. The wake-up interrupts are also part of ISR. Therefore, it has to be checked if one of the following interrupts is pending: CAN wake-up interrupt (CWI), 2x LIN wake-up interrupts (LWI1|LWI2) and 2x WAKE pin interrupt (WI1|WI2) (valid for UJA1078A). At the end of the wake-up related actions the pending wake-ups must be acknowledged by writing the related bit in the ISR. Moreover, the V1 undervoltage interrupt must be cleared because this interrupt is always pending when V1 has been enabled. It shows that the V1 supply was low, which is always the case in Sleep Mode.

If no power-on or wake-up can be recognized it must be checked if a more detailed diagnosis is possible. This is done by checking the configuration of the V1 undervoltage interrupt enable (V1UIE) in the MCR. If this bit is not set, it is impossible to determine between type 2 and type 3 reset events and therefore no detailed diagnosis is possible. But if the V1UIE bit is set, it is possible to check if the V1 undervoltage interrupt (V1UI) is

pending. A pending V1UI indicates a V1 undervoltage event or an overtemperature shutdown. The microcontroller supply was low in both cases (type 2). The V1UI interrupt is cleared by writing 0x6800 to ISR. If no V1UI is pending the only possible reset source are type 3 resets (watchdog related resets). When these kinds of resets are detected a re-initialization of the complete application is required because the root cause of the reset is unknown. It could be a corrupted stack or memory or e.g. corrupted SPI commands etc.

The last part of the reset and wake-up source detection is to check if other interrupts like Cyclic Interrupt (CI) and V2 Undervoltage Interrupt (V2UI) are pending. If they are pending they can be cleared by writing 0x6480 to ISR.

Background of these bits:

In case the first watchdog trigger is not performed within the first 128ms after reset, the CI bit is set. This indicates that there was an overflow of the Timeout Watchdog in Standby Mode. This indicates that the startup of the microcontroller takes too long. Therefore it has to be ensured during the software development process that the startup is done within 128ms.

V2UI can only be pending if the UJA107xA was in Normal Mode and a V2 failure occurred before the reset event. A V2 failure is no reset event. Therefore, the UJA107xA stays in Normal Mode, but the CAN Transceiver is disabled. During Startup Operation the V2UI requires no further actions because the CAN Transceiver is not in normal mode and the CAN Protocol Engine (PE) of the microcontroller is re-initialized.

3.1.4 Limp Home handling

The last step before entering Normal Mode is configuration of Limp Home Warning Control (LHWC) and Limp Home Control (LHC). Both bits can be cleared with the following write access to the Mode Control Register (MCR): 0x20xx. The last byte of this access is defined by the initialization of the UJA107xA.

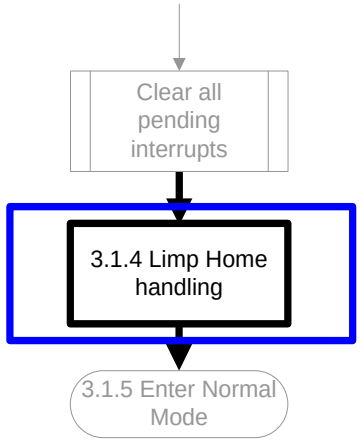


Fig 59. Limp Home handling

It is recommended to always clear LHC because the application is running properly at this point in time and therefore Limp Home can be disabled. If LHWC is cleared depends on the safety level of the application. If every system reset should trigger Limp Home immediately the application should set LHWC and always write 0x22xx to MCR. If two consecutive resets without software interaction should trigger Limp Home, it is recommended to clear LHWC. With LHWC cleared, a system reset will not directly activate Limp Home and give the application a chance to re-start properly. With this first reset event LHWC will be set and prepare the Limp Home event. If now a second reset occurs (e.g. because the software is seriously damaged) the Limp Home output of the SBC will be activated.

3.1.5 Transition to Normal Operation

Normal Mode is entered by writing e.g. 0x2Cxx to the Mode Control Register. For more information on Normal Operation refer to chapter 3.2.

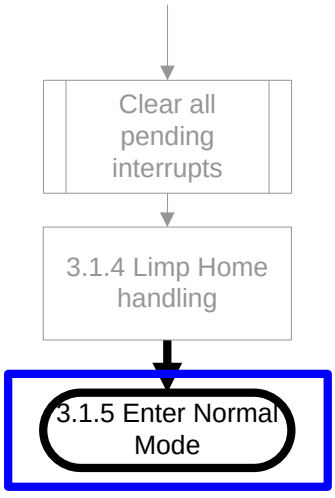


Fig 60. Transition to Normal Mode

3.2 Normal Operation

This section introduces the software operations which are associated to Normal Operation of the application. Normal Operation is related to the Normal Mode of the UJA107xA. This mode can only be entered after a successful startup operation. Therefore the beginning of Normal Operation is a transition from Standby to Normal Mode caused by the related SPI command.

The most important operations of the Normal Operation are the watchdog handling, the transceiver control and the execution of the application. For that reason this chapter shows how to realize successful watchdog triggers and to enable/disable the different CAN and LIN Transceivers. At the end of the Normal Operation a transition to Low Power Operation (Standby, Sleep) is performed via the related SPI command.

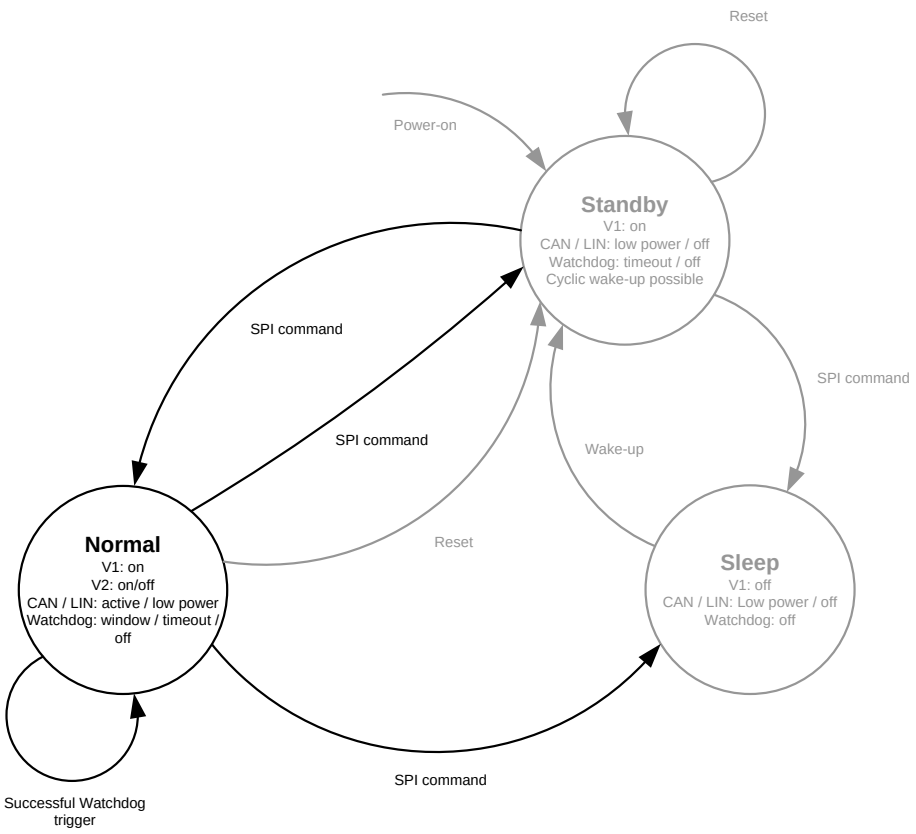


Fig 61. Simplified mode diagram with focus on Normal Operation

Fig 62 shows the different parts of Normal Operation. It consists of:

- Watchdog trigger
- Transceiver control
- Execution of application
- Transition to Low Power Operation (Standby, Sleep)

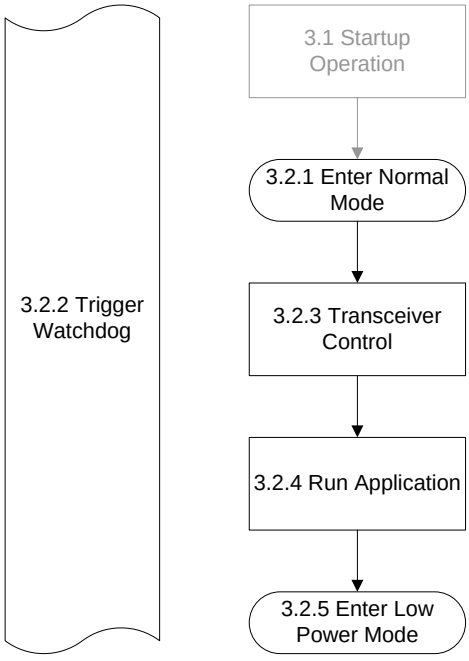


Fig 62. Normal operation software flow

3.2.1 Transition to Normal Operation

The transition to Normal Operation is done by changing the UJA107xA mode from Standby Mode to Normal Mode. This is done by writing 2 or 3 to the Mode Control bits of the Mode Control Register (MCR). If MC = 2, the internal V2 regulator is disabled and therefore, an external CAN Transceiver supply is required to supply the SBC internal CAN Transceiver (if available in the UJA107xA derivative).

In case of MC = 3, the internal V2 regulator is powered up to supply the SBC internal CAN Transceiver (if available) or other external hardware.

A transition to Normal Mode will not trigger the watchdog. Therefore it is possible to change the modes at any point in time.

In Normal Mode additional functionality of the SBC becomes available as there is now access to the EN pin behavior and the V2 undervoltage detection.

In Normal Mode the EN pin always goes high when the Enable Control (ENC) bit is set. This can be used to control safety critical hardware in the application. With any reset event, EN will go LOW and protects the connected external hardware from serious misbehavior. ENC remains unchanged it is controlled by the software.

A transition to Normal Operation can be e.g. performed by writing 0x2C80 to the MCR. This command will change to Normal Mode with enabled V2 and the EN pin is high as long as the UJA107xA is in Normal Mode and the ENC bit is set.

The second Normal Mode feature is the V2 undervoltage detection. The undervoltage detection mechanism will always work in Normal Mode, but the undervoltage interrupt is only triggered when it is enabled. It is recommended to always enable the V2 undervoltage interrupt. This is done by writing 0x4400 to the Interrupt Control Register (ICR).

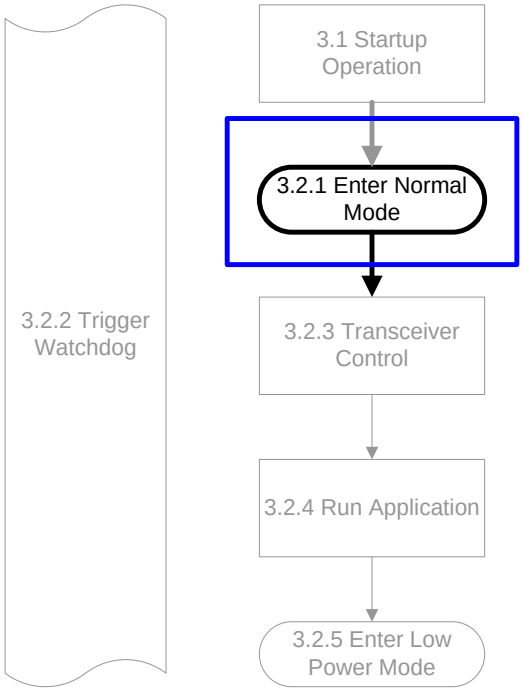


Fig 63. Transition to Normal Operation

3.2.2 Watchdog Trigger

Successful watchdog triggers on a regular basis are required to keep the UJA107xA in Normal Mode. This must be ensured by the application software. If the watchdog is enabled, it must always be triggered regardless of the UJA107xA Mode. Therefore, it is recommended to treat the Timeout Watchdog in Standby Mode like a Window Watchdog in Normal Mode. This makes the software architecture easier by just having a common watchdog service principle. So, there is no need to distinguish between the SBC

operating modes. Anyhow, in some applications it might be of benefit to have a relaxed watchdog trigger mechanism during Standby Mode. This depends on the overall software architecture used and is supported by the timeout behavior of the SBC in Standby Mode.

In general, it is not recommended to trigger the watchdog directly within a related timer interrupt service routine, because a hardware timer within a microcontroller might still operate properly, while the main microcontroller core is crashed already. Instead, it is recommended to install software flags which indicate proper operation of the application. This way, the main part of the watchdog trigger task is to check whether all expected flags are set. If all flags are set correctly the watchdog can be triggered with the related period. In case of incorrect flags a software reset can be performed or the watchdog trigger can be skipped. Both actions will end up in a reset as desired.

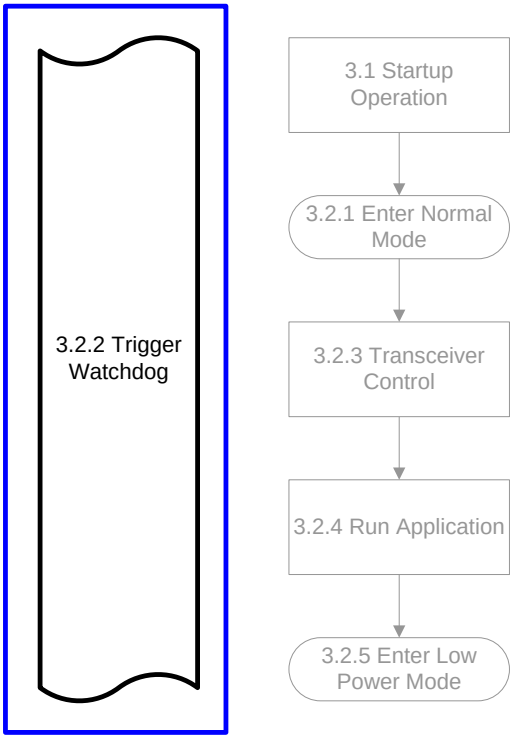


Fig 64. Watchdog Trigger

In Normal Mode the watchdog is running in Window or Timeout Mode. It is recommended to use the Window Watchdog Mode for Normal Operation.

The watchdog is in Window Mode when the Watchdog Mode Control (WMC) bit of the Watchdog and Status Register (WDSR) is set to 0 in Normal Mode. A watchdog trigger command for Window Watchdog Mode with e.g. 8ms Nominal Watchdog Period (NWP) is 0x0000. This code must be written to the WDSR within the open watchdog window (in

case of 8ms the open window is 4.4ms to 7.2ms). For more information on the watchdog refer to chapter 2.9.

In case of a detected software/application failure a software reset can be triggered. This is done by setting the SWR bit in the WDSR with the following SPI command: 0x0080. The reset is triggered when the chip select goes high again.

Moreover, a change of the WMC bit in Normal Mode causes a system reset. This is to make sure, that an unintended change of the Watchdog setting is detected by the system and the Watchdog can not be “disabled” on the fly without being noticed. Therefore, it is recommended to change to Standby Mode before changing the WMC bit, if this is desired.

3.2.3 Transceiver Control

The next step after the transition to Normal Mode is the control of the different UJA107xA transceivers. The transceiver control is done by the related Standby Control (STBxC) bits in the Interrupt Control Register (ICR). The enabling of the transceivers is done by clearing the related STBxC bits (STBCC, STBLxC).

Before starting CAN communication, or rather enabling the CAN Transceiver and the CAN PE it is recommended to check if the CAN Transceiver is already supplied. Hence, the V2 Status (V2S) bit in the WDSR should be read. Only if WDSR_V2S = 0, the CAN transceiver should be activated. If STBCC has been cleared before, the CAN Transceiver is not active until WDSR_V2S = 0. Only if it is ensured that the CAN transceiver is active the CAN PE should be activated and CAN communication can be started. If this sequence is not considered, active sending on CAN may lead to error messages on the bus.

In general it is recommended to disable all wake-up sources while in Normal Mode. Benefit of this is, that any unintended exit out of Normal Mode would immediately cause a system reset. This is to prevent a dead-lock situation of the system with having all wake-up sources disabled and leaving normal operating mode.

Therefore, it is recommended to also clear the WICx bits that disable the local wake-up interrupts. This is done by a write access to the ICR with the following SPI command: 0x4C04. Within this command V1UIE and V2UIE are still set and the 70% reset threshold is selected. Hence, the result of this SPI command is that all wake-up sources are disabled and the failure detection features are still enabled.

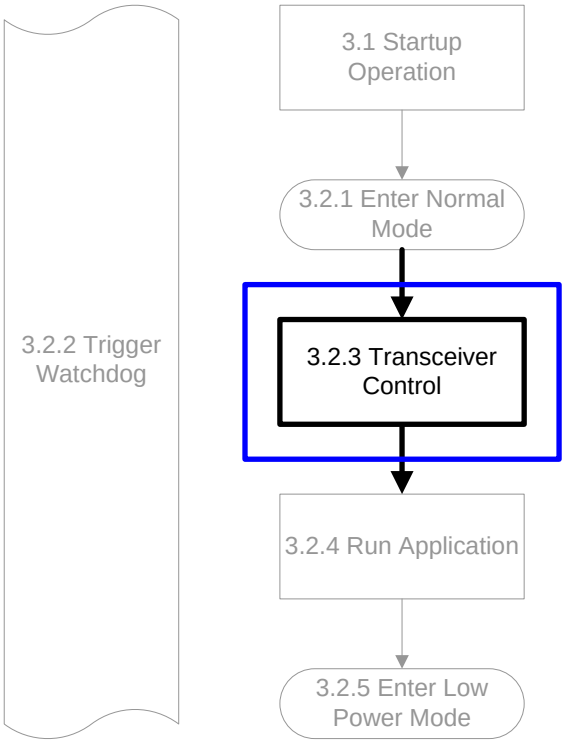


Fig 65. Transceiver Control

Any system reset will automatically disable all UJA107xA transceivers. Moreover, a V2 undervoltage (V2 < V_{uvd}) disables the CAN Transceiver while a battery undervoltage (V_{BAT} < $V_{uvd(LIN)}$) disables the LIN Transceiver. Both CAN and LIN Transceivers will automatically recover when the undervoltage is gone.

The V2 Undervoltage Interrupt (V2UI) can be used to control the CAN Protocol Engine (PE) within the microcontroller in case of a transceiver supply failure. This is illustrated by the figure below.

The V2 Status (V2S) bit in the WDSR indicates whether V2 is still low. If WDSR_V2S = 0 again, the CAN communication can continue because the transceiver is active again. On the other hand if WDSR_V2S = 1, it is recommended to disabled the CAN PE or at least stop transmitting CAN frames because the CAN Transceiver is disabled. Active sending on CAN with WDSR_V2S = 1 will lead to an increase of the CAN PE failure counters. The CAN PE can be re-enabled or data can be transmitted when WDSR_V2S is set to 0 again. It is recommended to poll the WDSR_V2S with a timeout of several ms to prevent a lock of the application in case of a long/permanent V2 undervoltage.

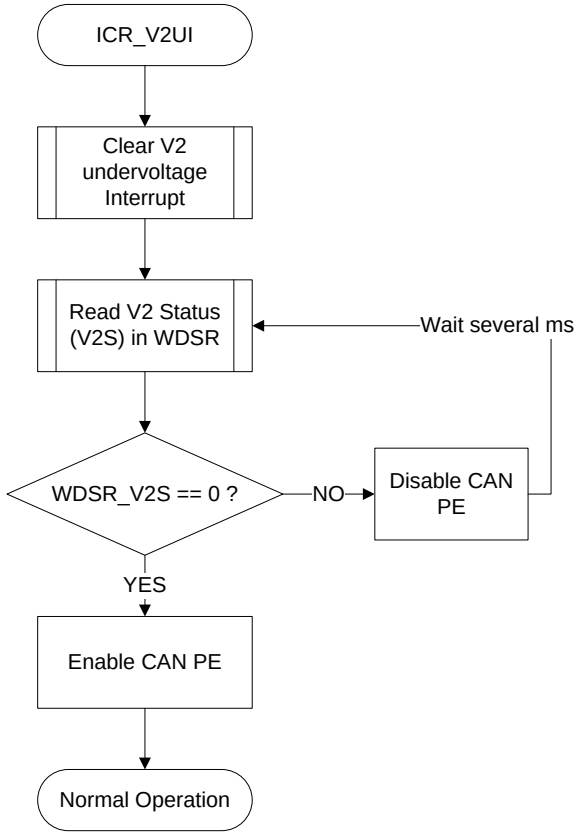


Fig 66. V2 failure operation

Another benefit of the UJA107xA family transceiver control mechanism is the individual transceiver control. Therefore, it is possible to treat each UJA107xA transceiver as a standalone one from software perspective that allows to reuse AUTOSAR Basic Software Transceiver Drivers.

3.2.4 Run Application

This is the most important part of the software flow and highly depends on the dedicated application. Therefore this chapter only gives an example of what is possible. The execution of the application can e.g. include CAN and LIN message routing, periodic input pin polling, battery supervision. From UJA107xA perspective it is important to keep the UJA107xA in Normal Mode with the related watchdog triggers and stop the CAN communication in case of a V2 undervoltage. Therefore the user of the UJA107xA has the freedom to adopt the software to its individual needs. Transition to Low Power Operation like Standby and Sleep Mode is always possible at any point in time regardless of the actual watchdog expiration time.

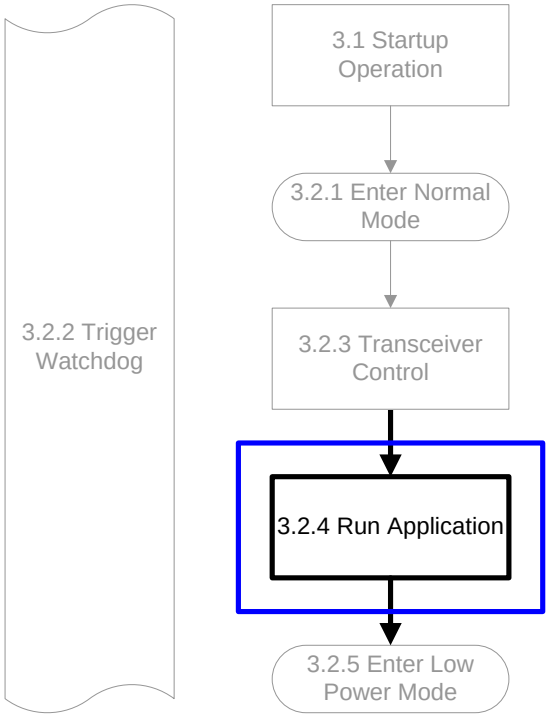


Fig 67. Run Application

3.2.5 Transition to Low Power Operation

Low Power Mode is entered by writing e.g 0x24xx or 0x20xx to the Mode Control Register (MCR). It is important to enable at least on wake-up source before entering SBC Standby or Sleep Mode. Without having any wake-up source active (CAN / LIN / local WAKE) the SBC will perform a reset in order to prevent a dead-lock situation (Sleep Mode forever). For more information on Low Power Operations refer to chapter 3.3.

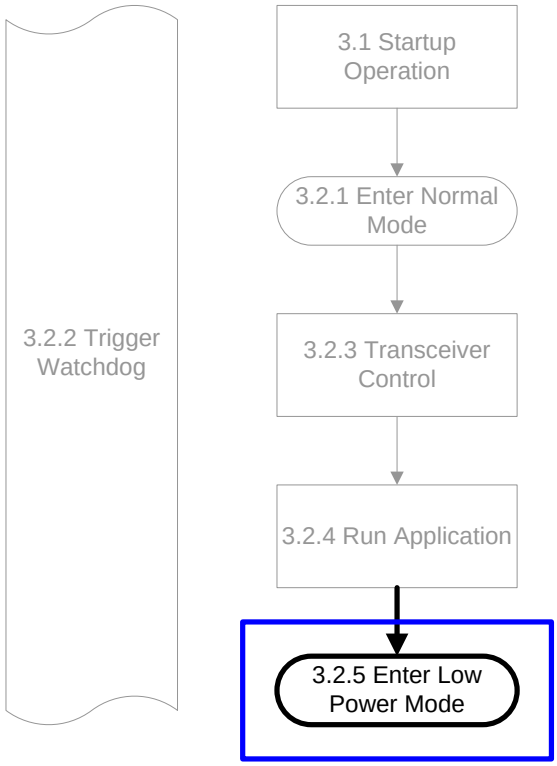


Fig 68. Transition to Low Power Operation

3.3 Low Power Operation

This section introduces the software operations which are associated to low power operation of the application. It is related to the Standby and Sleep Mode of UJA107xA. The difference between both modes is the amount of power that can be saved. In Sleep Mode, there is the lowest current consumption but the delay between wake-up event and restart of application is longer as in comparison to the Standby Mode, because the system has to start from an un-powered situation. In Standby Mode the current consumption is slightly higher compared to Sleep mode with the benefit of a much shorter delay between wake-up event and re-start of application. Therefore the following subchapters discuss the Standby and Sleep Mode in more details.

As illustrated in Fig 69, Low Power Operation can be entered directly from Normal Mode via SPI command. Moreover, Sleep Mode can also be entered via a temporary transition to Standby Mode by using two consecutive SPI commands (1. enter Standby Mode, 2. enter Sleep Mode). All mode transitions are performed by the dedicated SPI commands.

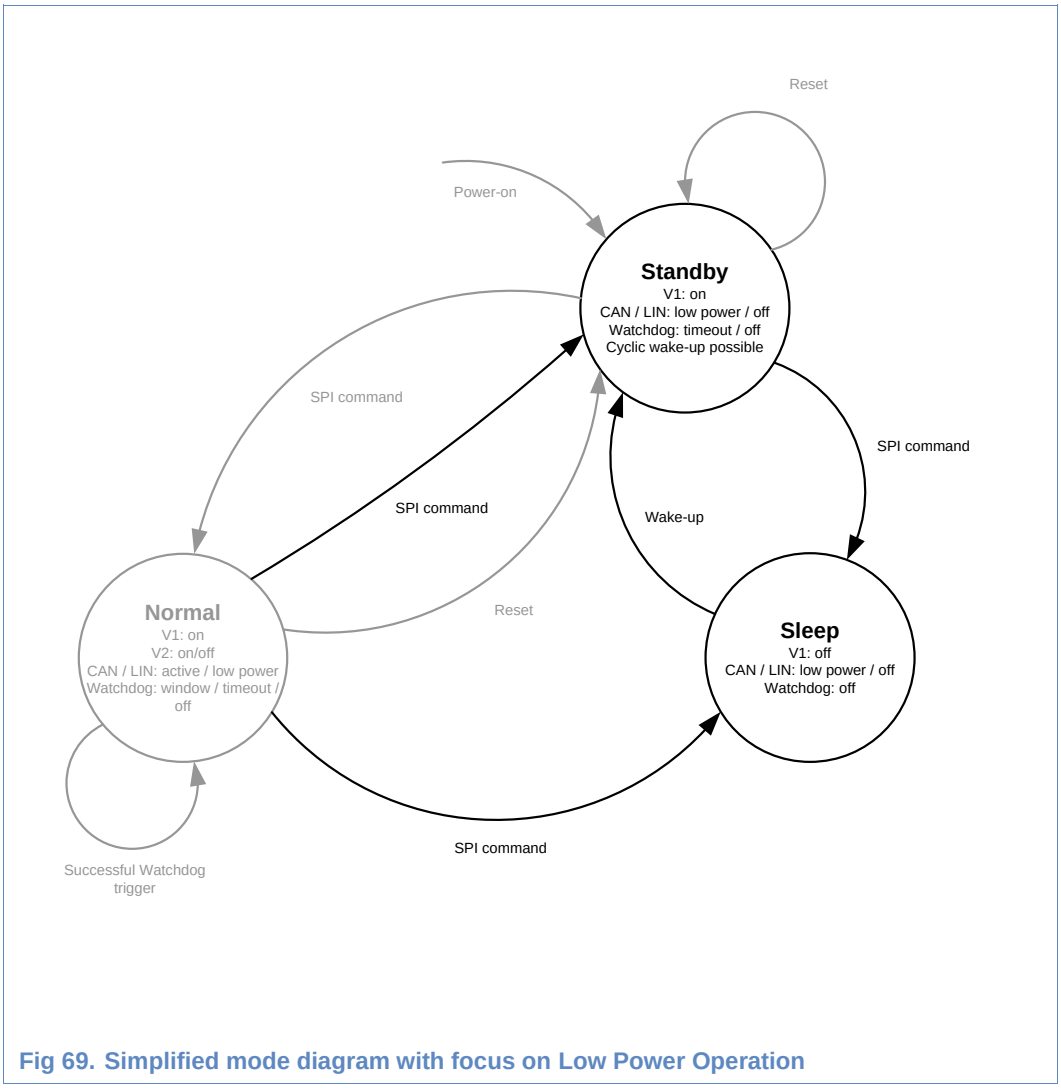


Fig 69. Simplified mode diagram with focus on Low Power Operation

3.3.1 Interrupt Handling

The different interrupts in the Interrupt Status Register (ISR) can be used as a wake-up event from Low Power Operation. All interrupts except of V2UI and POSI can be used as wake-up out of Standby Mode whereas CWI, WIx and LWIx can also be used as wake-up out of Sleep Mode. The “interrupt handling” in case of wake-up out of Sleep is performed by the Startup Operation because it is linked to a system reset event with powering up the application (cp. chapter 3.1). The only difference is the trigger event of the Startup Operation, which is now an interrupt and not a reset.

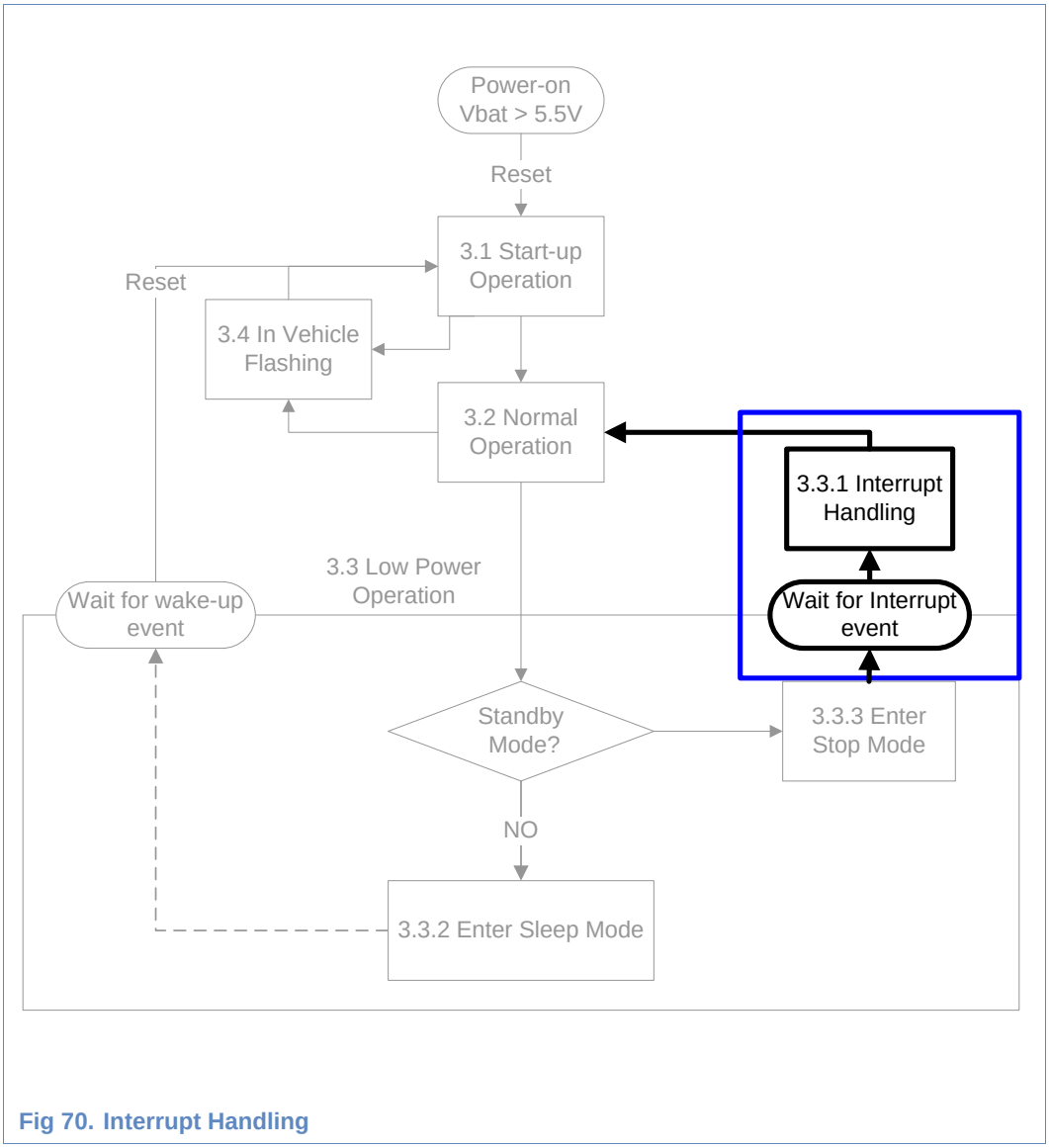


Fig 70. Interrupt Handling

The figure below illustrates the software flow of the interrupt handling. When an interrupt is pending the INTN pin goes low. The interrupt service can be implemented with an external interrupt to the microcontroller through the INTN pin that is triggered with a

falling edge or by reading the Interrupt Status Register (ISR) on a regular basis (polling). After a read access to the ISR the read data are evaluated. If any interrupt is pending (e.g ISR_V2UI) the related interrupt handler is called. The interrupt handlers are application specific functions and depend on the need of the application. In case of an ISR_V2UI interrupt handler the CAN PE is recommended to be disabled according to Fig 66. The interrupt handling loop is repeated as long as interrupts are pending. Moreover, INTN is low as long as at least one interrupt is pending. After each interrupt loop a short timeout should be implemented to prevent a lock of the application.

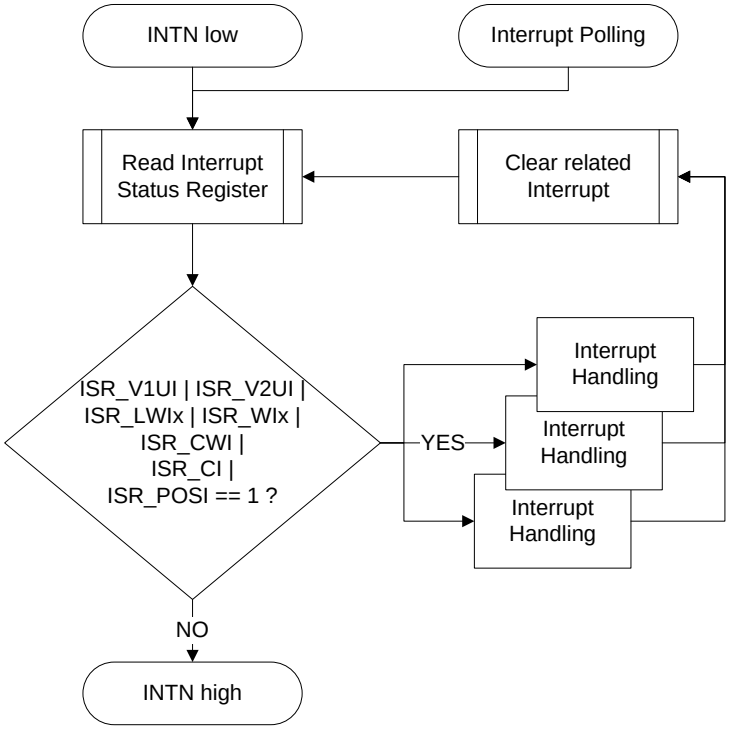


Fig 71. Interrupt Handling flow chart

3.3.2 Sleep Operation

Sleep Mode is a special kind of Low Power Operation. It can be entered from Standby and Normal Mode via the 0x24xx SPI command, which is a write access the Mode Control Register (MCR). There are only few possibilities to leave Sleep Mode and enter Standby Mode: There must be a wake-up event on CAN, LIN or the WAKE pins or there has to be a power-on event on the pin BAT. After the wake-up event the startup operation is performed. For more information on startup operations refer to chapter 3.1.

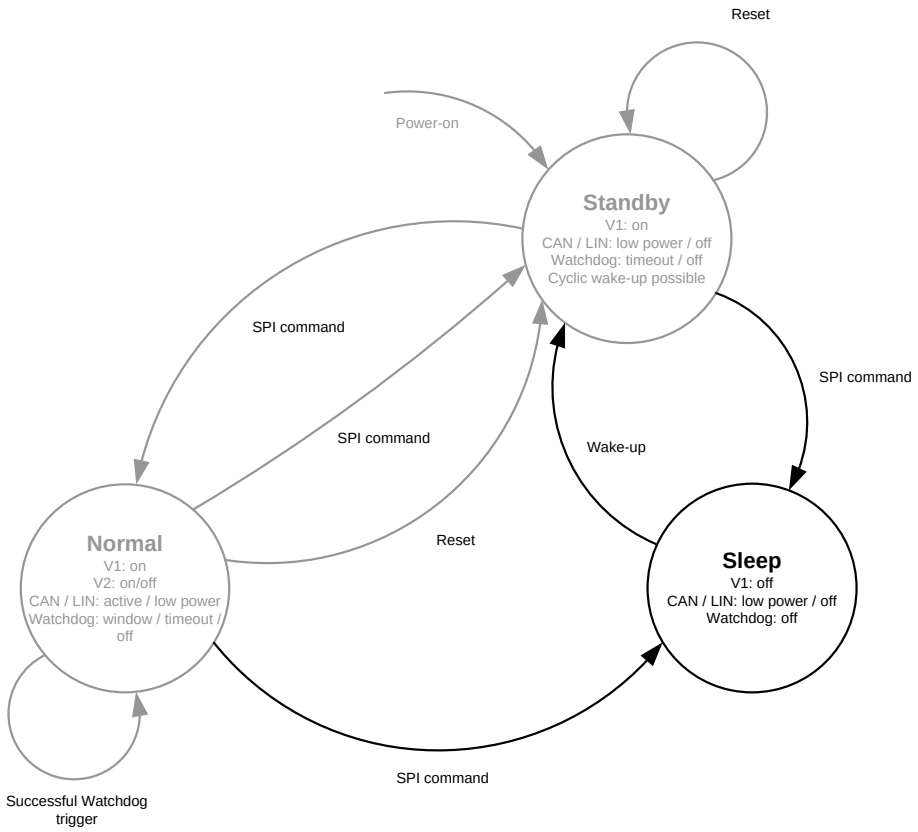


Fig 72. Simplified mode diagram with focus on Low Power Operation (Sleep Mode)

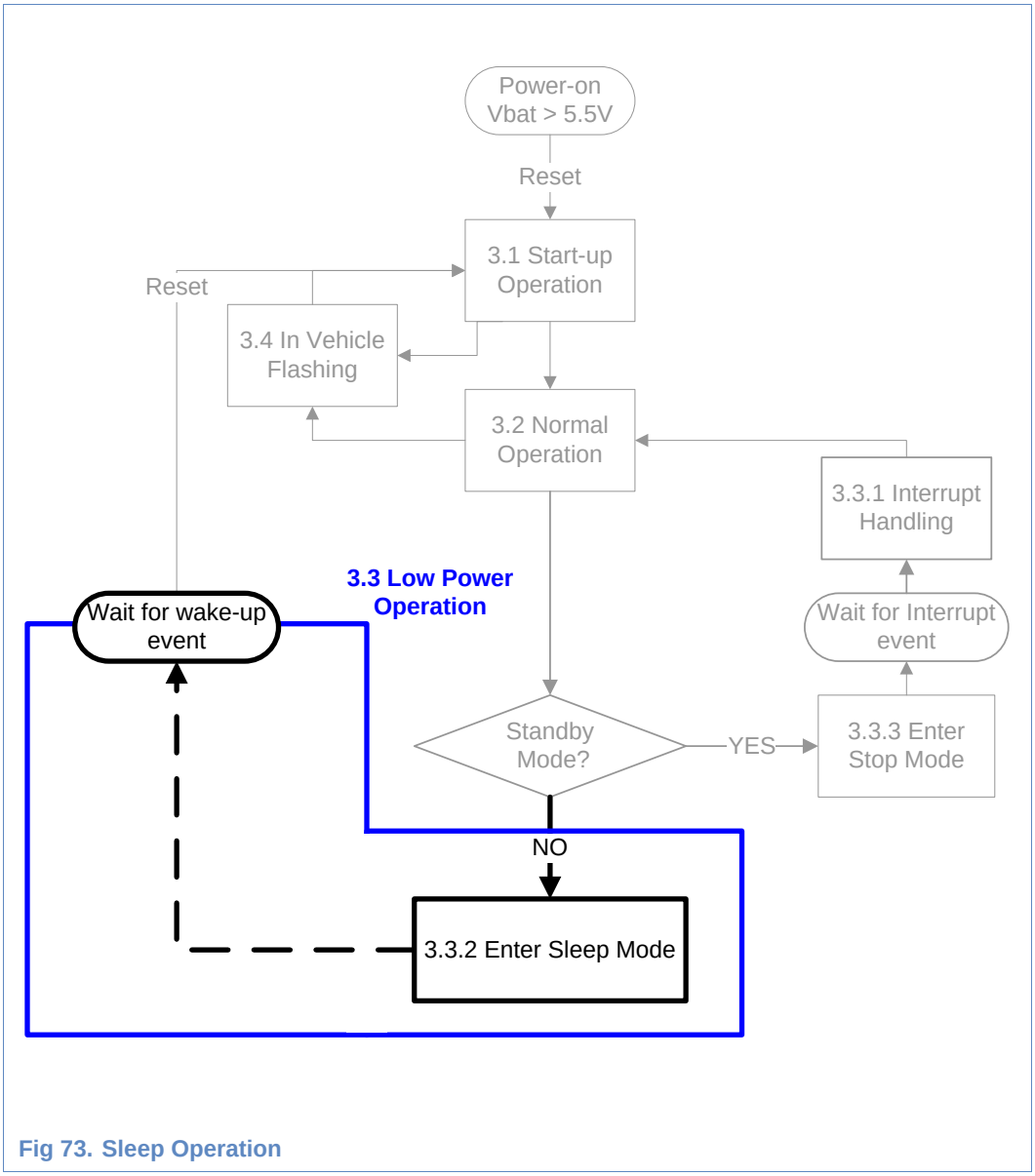


Fig 73. Sleep Operation

The software flow for entering Sleep Mode is shown in Fig 74. The first action of the goto sleep routine is the check for enabled wake-up sources. Entering Sleep Mode without any active wake-up source will end up into a reset in order to prevent a possible dead-lock without any wake-up activated. Therefore at least one wake-up source must be enabled.

Via the Interrupt Control Register (ICR) the wake-up sources can be activated. The 0x4FFF command for example enables all wake-up sources, the V1 and V2 undervoltage interrupt, the sampling of the WAKE pins and sets the reset threshold to 70%.

Having the wake-up sources configured properly, a read access to the Interrupt Status Register (ISR) must be performed (0x7000). If an interrupt is pending the UJA107xA cannot enter Sleep Mode and any attempt to enter Sleep Mode would lead to a system

reset. Therefore, the goto sleep process must be aborted and the related interrupt handler must be called. If no interrupt is pending the UJA107xA can successfully enter Sleep Mode with a 0x24XX write access to MCR. It is recommended to always clear MCR_LHWC before entering Sleep Mode because otherwise the wake-up event will trigger the Limp Home circuitry (Note: Any reset with LHWC set causes the LIMP output to become active and since V1 of the microcontroller is un-powered in Sleep Mode, there is a reset with waking-up from Sleep Mode).

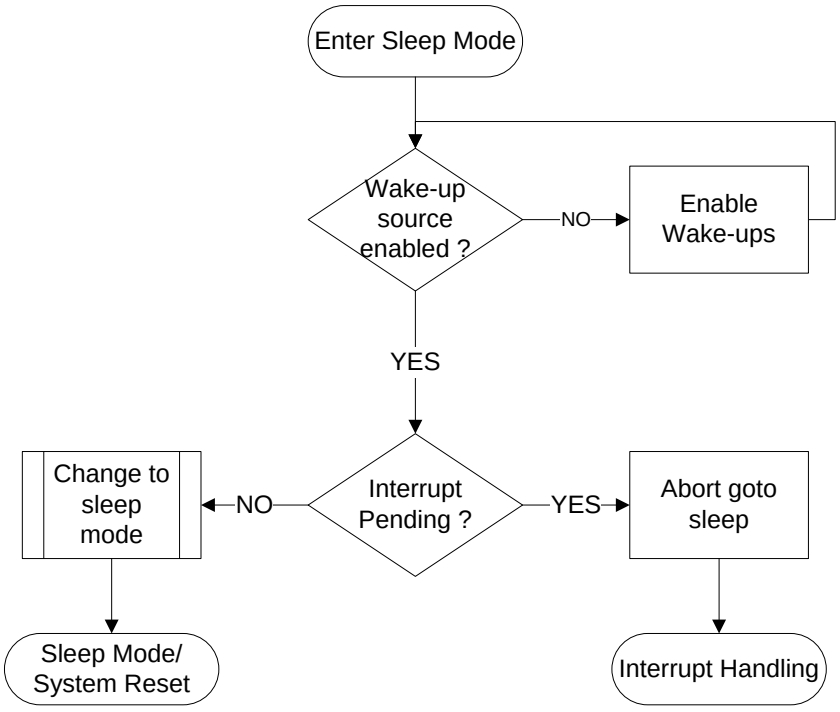


Fig 74. Enter Sleep Mode flow chart

3.3.3 Standby Operation

A different flavor of Low Power Operation is the so called Stop or Sub Clock operation. This operation is related to the Standby Mode of the UJA107xA because it requires a supplied microcontroller. Stop or Sub Clock operation is a low power feature of the microcontroller itself. Therefore, it will take place in Standby Mode of the UJA107xA and a wake-up from Stop/Sub Clock operation will not lead to mode changes of the UJA107xA.

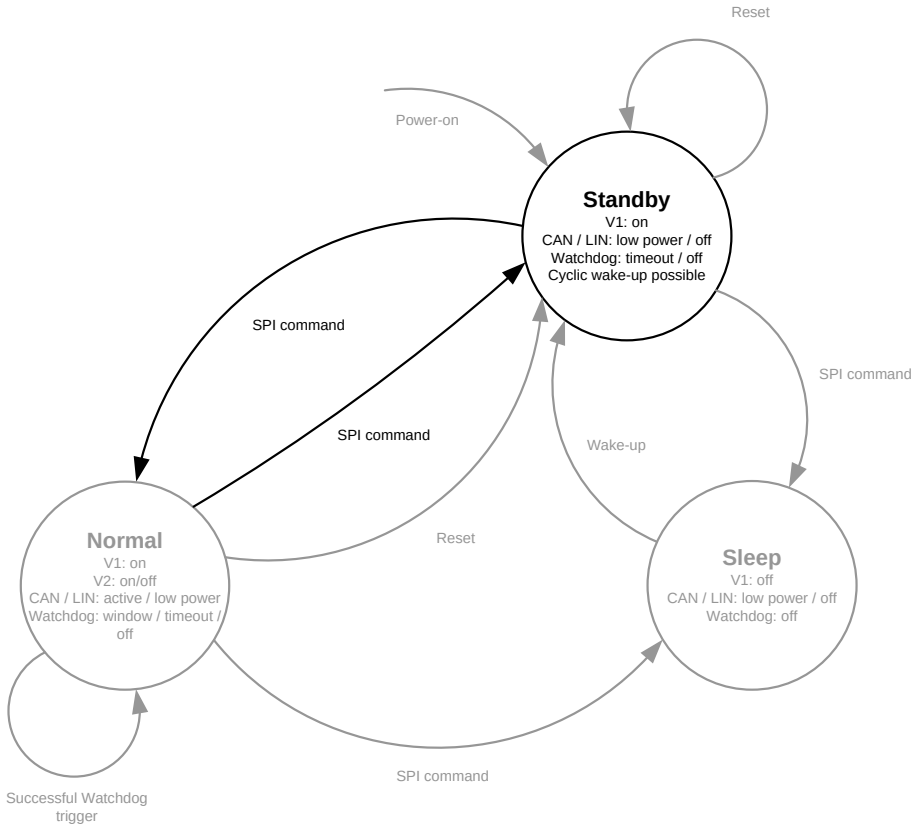


Fig 75. Simplified mode diagram with focus on Low Power Operation (Stop/Sub Clock)

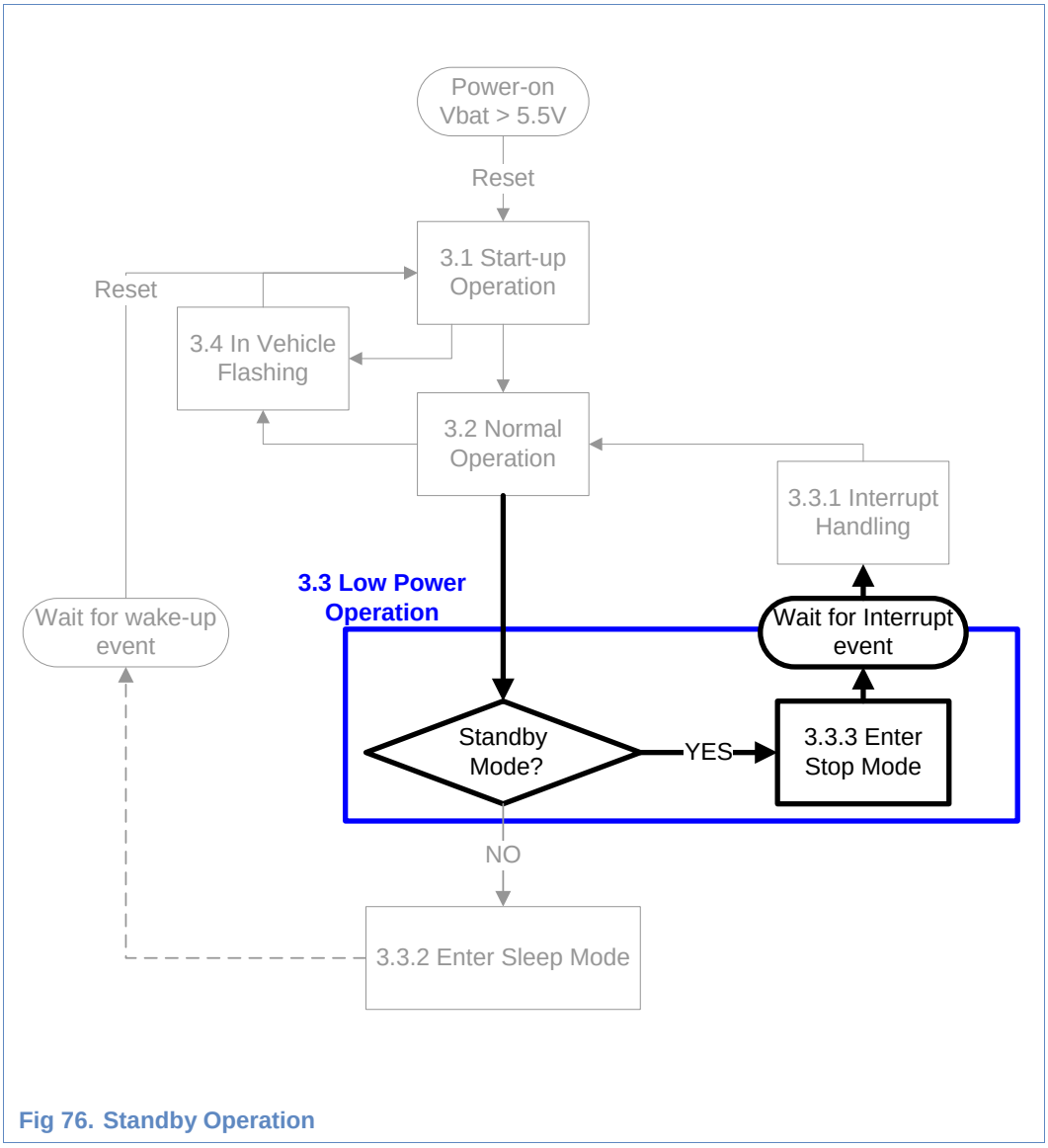


Fig 76. Standby Operation

Stop or Sub clock operation can be realized by two different ways that are discussed within the next subchapter. One part is the pure Stop operation and the other one is the cyclic wake-up out of Stop operation

3.3.3.1 Stop/Sub Clock Mode

The first part of the Stop operation is the transition to Standby Mode with the SPI write access 0x20XX to the Mode Control Register (MCR). Afterwards it is checked if the necessary wake-up sources are enabled. If not they are enabled accordingly. Fig 77 shows the required actions before entering Stop Mode. When the interrupt sources are configured properly the software can check if interrupts are still pending. This is done by a read access to the ISR (0x7000). If an interrupt is pending for safety reasons the watchdog of the UJA107xA cannot be disabled and therefore no Stop Mode of the

microcontroller is possible. As a consequence the goto stop process must be aborted and the related interrupt handler must be called. If no interrupt is pending, the watchdog can be disabled by setting the Watchdog Mode Control (WMC) bit in the Watchdog and Status Register (WDSR). Any interrupt will automatically re-enable the watchdog. When the watchdog is disabled the Stop/Sub Clock operation of the microcontroller can be entered by disabling the oscillator completely or switching to a lower clock frequency.

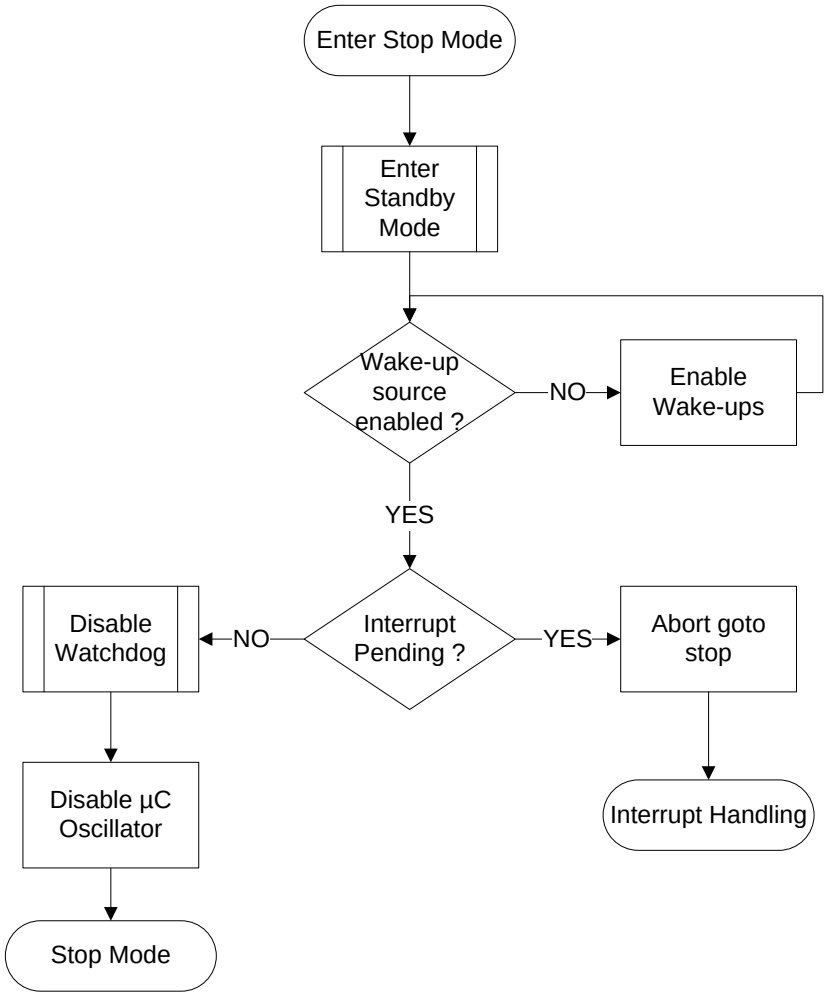


Fig 77. Enter stop/sub clock mode flow chart

3.3.3.2 Cyclic Wake out of Stop/Sub Clock Mode

The software flow of entering Stop Mode with cyclic wake-up is similar to that one introduced in the previous subchapter about pure Stop Mode. The only difference is the configuration of the watchdog. Here the watchdog continues to run while the

microcontroller stops operation. The idea is that an overflowing watchdog wakes-up the microcontroller cyclically depending on the selected watchdog period time.

Therefore, the UJA107xA must be in Standby Mode, the related interrupts must be enabled and no interrupt must be pending. If all these conditions are fulfilled the UJA107xA watchdog must still be enabled while disabling the microcontroller's oscillator. The watchdog is enabled in Standby Mode by e.g. writing 0x0700 to the Watchdog and Status Register (WDSR). The previous SPI command enables the Timeout Watchdog and hence, the Cyclic Wake feature with a period of 4096ms because WDSR_NWP = 7. The watchdog overflow (after 4096ms) triggers the CI bit and wakes up the microcontroller out of stop mode with the falling edge on the INTN pin. The release of the CI bit should not be done within the interrupt service routine because the interrupt mechanism could still work when the rest of the application is broken. Therefore it is recommended to handle the CI within the application.

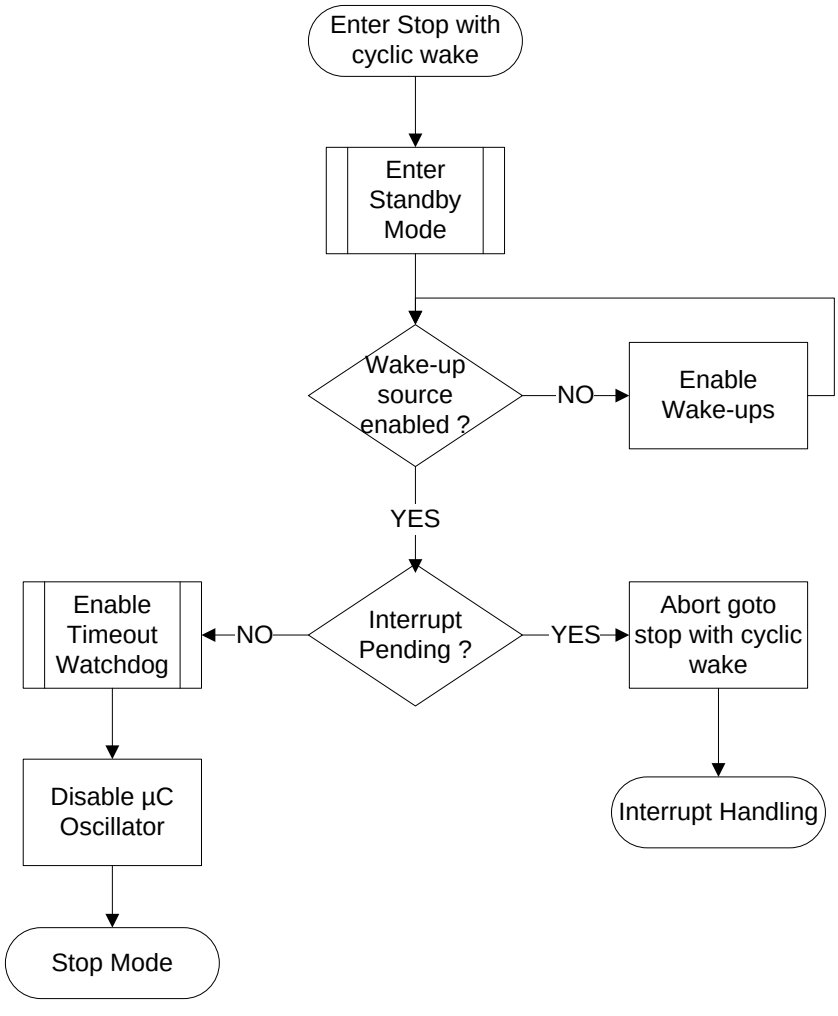


Fig 78. Enter stop with cyclic wake flow chart

3.3.4 How to avoid unintentional Sleep mode

When an application is not using Sleep mode and needs to be sure that Sleep mode cannot be entered unintentionally, this can be supported by HW and SW:

- Ensuring error-free SPI communication by proper signal routing on the PCB (see section 2.3.4)
- Detecting SPI communication errors (section 3.3.4.1)
- Using shadow registers (section 3.3.4.2)
- Disabling sleep mode by disabling all wake sources and/or by keeping at least one interrupt pending (section 3.3.4.3 and 3.3.4.4)
- Entering and leaving Standby mode with a sequence that keeps sleep mode disabled while SPI transfers are made (section 3.3.4.5)

“Disabled Sleep mode” means that an SPI command requesting sleep mode is not executed and instead a system reset is done.

3.3.4.1 Detecting SPI communication errors

It is possible to detect many of the SPI communication failures, which are caused by noise, shorts or line interruptions.

The address bits and the read-only bit of the SDI bit stream are always mirrored to the SDO bit stream so any difference between the two indicates an error. Therefore after each SPI access a simple comparison of the sent address/read-only bits with the echoed bits would detect errors, especially “stuck-at” errors, where one signal line has always the same value, because of a short or because of a signal interruption.

Corrupted data sent from µC to SBC with an SPI write access can be detected by adding an SPI read access afterwards, which reads back the data that has been written before and compares the two. For this comparison the read-only bit needs to be masked out, because this is intentionally different.

These checks can be implemented in an SPI access function like illustrated with the pseudo code below:

```
SpiSendAndCheck(pointer to data)
    ReadOnlyMask = 0x1000  // marks the location of the read-only bit
    AddrRoMask = 0xf000    // marks the address bits and the read-only bit
    tx_data = data         // remember what shall be sent
    SpiSend(pointer to data)  // send data, received data will be stored in same variable,
                           //    no matter if this was a write or a read-only access
    rx_data_1 = data       // remember the received data
    // check mirrored address bits and read-only bit
    if ( (rx_data_1 & AddrRoMask) == (tx_data & AddrRoMask) )
        SPI_error_detected = false
    else
        SPI_error_detected = true
    // if it was a write access, read back the data written previously and compare
    if ( (tx_data & ReadOnlyMask) == 0 )    // if it was a write access...
```

```
data = tx_data | ReadOnlyMask // ...change it to “read-only”...
SpiSend(pointer to data)   // ...and repeat SPI access, same data, but read-only
rx_data_2 = data           // this is the register contents after the second SPI access
                           // it should echo the written data
// compare
if (rx_data_2 != (tx_data | ReadOnlyMask) )
    SPI_error_detected = true
data = rx_data_1     // in case it was a read access, this is the read data
```

Randomly corrupted data sent from the SBC to the µC could be detected by repeating the SPI read access and thus reading the data twice and comparing the results. However, if both results are not the same, it may not be clear if there was actually an SPI error, or if the content of the register has actually changed between the two read accesses. The latter can apply to all those register bits that can be changed by the SBC. Therefore this check is not very practical and it has not been included in above pseudo code example.

3.3.4.2 Using shadow registers

In case that SPI data corruption cannot be excluded by HW design, one should consider that reading data from the SBC may not always be reliable. Then read-modify-write accesses shall not be done directly to SBC registers, but only to variables in the µC, which serve as shadow registers. This means that the SW needs to define for each SBC register one variable. The SW first changes the content of one of these variables (i.e. of one shadow register) and then transfers that content of the variable to the corresponding register in the SBC via SPI.

Example: The ENC bit in the Mode_Control register shall be cleared, but all other bits shall keep their old value.

Let’s look at two options how to do this:

A) Without shadow register:
   Read the Mode_Control register from the SBC
   Clear the ENC bit of the read data
   Write the modified data to the Mode_Control register of the SBC

B) With shadow register:
   Clear the ENC bit of the shadow register of the Mode_Control register
   Write the modified shadow register to the Mode_Control register

With option A) any SPI error during reading would be written back to the SBC and this way unintended commands could be written. With option B this cannot happen, because the modified register content is not based on a previous SPI read access.

3.3.4.3 Disabling all wake sources

When no wake source is enabled, sleep mode is disabled. When CAN and LIN are used for communication, their wake-up sources have to be disabled anyway (STBCC=STBLCx=0) in order to get these transceivers into active mode.

In order to disable the WAKE pin wake source, both WAKE pins have to be disabled (WICx=0). However, then it is not even possible to read the value of the signals at the

WAKE pins. If these signals must be readable, that wake source has to be enabled. In this case Sleep mode can still be disabled by keeping interrupts pending (section 3.3.4.4).

A worst case scenario, which has to be avoided, looks like this:

The SBC is in Normal mode, all available CAN and LIN transceivers are used for communication, no interrupts pending, WAKE pins are used to read the signals connected to these pins, **but** these signals are **not** usable to wake-up the SBC from Sleep mode. With this configuration a corrupted, unintended SPI command that looks like a sleep mode request causes the SBC to enter sleep mode. Then the only wake-up possibility would be a power-on reset, because CAN and LIN wake-ups have not been enabled before and the only active wake sources are the WAKE pins, which however are not connected to appropriate wake signals.

3.3.4.4 Keeping interrupts pending

While an interrupt is pending (V1UI =1 or V2UI =1 or CWI=1 or LWIx = or WIx=1 or POSI=1 or CI =1), Sleep mode is disabled. Right after power-up the POSI interrupt gets active. By never clearing this interrupt, Sleep mode is disabled all the time.

A pending interrupt can be created while the SBC is in Standby mode, by letting the watchdog overflow once, which sets the CI interrupt.

3.3.4.5 Entering and leaving standby mode

When a module shall be put into standby mode, usually at least one wake source in the SBC needs to be enabled and all interrupts need to be cleared in order to be ready for wake-up from standby. At first glance this may look like a potential risk for unintended sleep mode, since in the sections above it was said that either all wake sources should be disabled or at least one interrupt should be kept pending in order to exclude unintended sleep mode. However, with an appropriate sequence of SPI commands it can be ensured that sleep mode stays disabled until the last SPI command has been executed. Then it is not necessary anymore to disable sleep mode, because there are no further SPI commands that could be corrupted in a way that they cause unintended sleep mode. The recommended sequence is listed in Table 16.

Table 16. Recommended sequence for putting a module into standby mode

Step #	Step	Sleep mode disabled by…
0	SBC is still in normal mode	No wake sources enabled and/or interrupt(s) pending
1	Set SBC to standby mode	No wake sources enabled and/or interrupt(s) pending
2	Disable all wake sources	No wake sources enabled. Eventually also interrupt(s) pending
3	Clear all interrupts	No wake sources enabled.
4	If no cyclic wake-up is wanted, disable watchdog	No wake sources enabled.
5	Enable desired wake source(s).	After this SPI command sleep mode is no longer disabled. However, there will be no further SPI commands that could cause sleep mode.

In case that the WAKE pins are used as relevant wake source, the sequence above needs to be extended with a step #6, which is a check of the WAKE signals. This is necessary in order to exclude that these signals are already at the level that is associated with a wake-up. For example, if one of the WAKE pins is connected to terminal 15 (ignition signal), it needs to be checked if this signal is actually low and waiting for a rising edge. This means that an additional SPI transfer is necessary, although Sleep mode is not disabled anymore. However, since the necessary bit pattern for reading the WAKE pins is

0001 xxxx xxxx xxxx,

the bit pattern

00**01 10**xx xxxx xxxx

can be used, which has a maximized Hamming Distance of 4 compared to the relevant part of a bit pattern that would cause the SBC to enter sleep mode, which is:

00**10 01**xx xxxx xxxx.

This way the probability for unintended sleep mode because of this SPI access is very low.

Note: if the SPI command executed in step #5 is done with an SPI write access followed by an SPI read-access verifying the data written before (like suggested for error checking in section 3.3.4.1), then the pattern

0**101 10**xx xxxx xxxx.

should be used for the read access, in order to get the maximum Hamming distance, which is 5 in this case.

3.4 In Vehicle Flash Operation

This chapter will not explain the H.I.S Standard [4][5][6], how to flash a device within a vehicle. It only shows, how to put the UJA107xA into a relaxed watchdog mode, which allows watchdog services and flash programming in parallel.

The advantage of the timeout watchdog is that the trigger period is 2 times the Nominal Watchdog Period (NWP). Therefore, the longest period in Timeout Watchdog Mode is 2x4096ms. This time is sufficient even for slow Flash memories with long sector erase times. The figure below shows the flow chart of the In-Vehicle Flash operation according to H.I.S. The typical starting point of the Flash Operation is the reception of the flash request via CAN in Normal Operation, which usually sets corresponding user defined control bits in the memory of the system or changes the flash signature in the Flash Memory Controller (FMC) of the microcontroller.

The first step of the software flow is the preparation of the programming process. After that process a system reset is requested. This can easily be done by changing the Watchdog Mode Control (WMC) bit in UJA107xA Normal Mode. A change of the WMC bit in Normal Mode immediately forces a hardware reset.

After the reset the Startup Operation is not entered because of the request for a flash update (user defined control bits or an evaluation of the flash signature cause this branch in the software flow). Now step 2 takes place. Before entering Normal Mode it has to be ensured that no interrupt is pending and the WMC bit in the Watchdog and Status Register (WDSR) is equal to 1 (relaxed watchdog triggering). If this is not the case a transition to Normal Mode will change the watchdog to Window Mode and this is not the intended behavior during flashing the application.

When the UJA107xA is in Normal Mode with Timeout Watchdog the CAN Transceiver needs to be enabled by clearing the STBCC bit in the Interrupt Control Register (ICR). As long as the watchdog is triggered within a time that is shorter than 2xNWP (no window required to be met), the UJA107xA will stay in Normal Mode.

If the complete data packages are flashed, the software consistency is checked and the user defined control bits are refreshed accordingly, the third step is executed. This requires an additional reset that can be done by changing the WMC bit back to the original setting. After that the normal Startup Operation starts again without entering the flash routines.

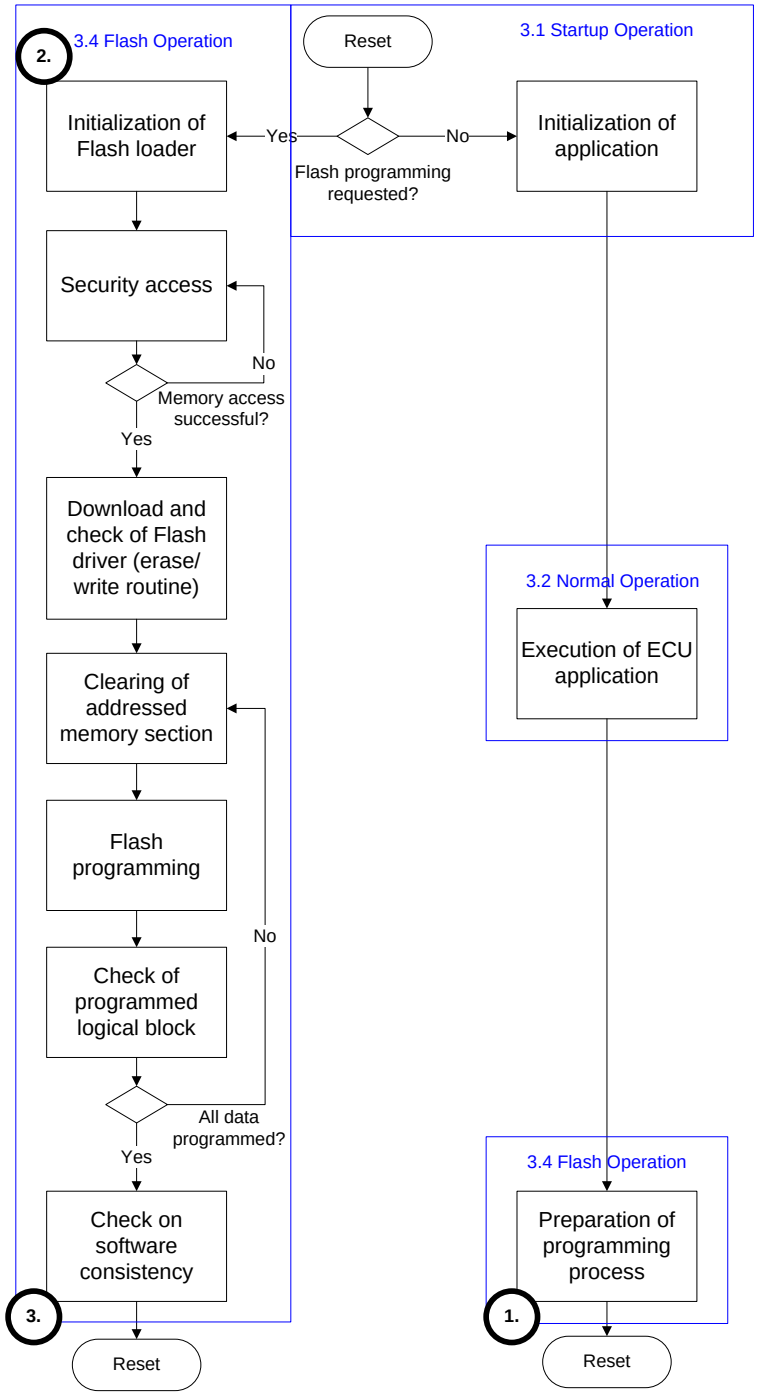


Fig 79. In Vehicle Flash operation flow chart

3.5 Software Development

For software development support UJA107xA provides the WDOFF pin. If this pin is high the watchdog timer of the UJA107xA is disabled and no watchdog resets are possible. Moreover, no cyclic wake is available. All other UJA107xA behavior is not influenced by the WDOFF pin functionality.

The status of the WDOFF can be identified by software by reading the Watchdog and Status Register (WDSR: 0x1000). The WDSR_WOS bit indicates the level of the WDOFF pin. If the read data of the SPI command is e.g. 0x1080 the WDOFF is high and therefore, the watchdog is disabled.

The WDOFF pin can be changed at any point in time. Each change will cause a reset in order to prevent an unwanted / not noticed change in the watchdog behavior. After the reset the watchdog is enabled or disabled according to the WDOFF level.

4. References

[1] UJA1078A High-speed CAN/dual LIN core system basis chip, Rev. 2 — 28 January 2011, Product data sheet

[2] LIN Specification Package, LIN Protocol Specification, Rev. 2.2A, LIN Consortium, 2010

[3] International Standard ISO 9141, Road Vehicles – Diagnostic Systems – Requirement for Interchange of Digital Information, International Standardization Organization, 1989

[4] HIS: Flashloader Specification v1.1, http://www.automotive-his.de/, January 25, 2007

[5] HIS: AK Flashbare Software – Ergebnisse, http://www.automotive-his.de/, January 25, 2007

[6] HIS: Functional Specification of a Flash Driver v1.3, http://www.automotive-his.de/, January 25, 2007

[7] Hardware Requirements for LIN, CAN and FlexRay Interfaces in Automotive Applications – AUDI, BMW, Daimler, Posche, Volkswagen – Revision 1.1 /2009-12-02

[8] PCA9703 - 18 V tolerant SPI 16-bit GPI with maskable INT, Rev. 2 — 14 June 2012, Product data sheet

[9] IBEE Ingenieurbüro für industrielle Elektrotechnik/Elektronik Prof. Sperling EMC Test report Nr. 09-04-10, May 05, 2010

[10] International Standard ISO 16750-2, Road Vehicles - Environmental conditions ad testing for electrical and electronic equipment: Part 2: Electrical transient conduction along supply lines only, International Standardization Organization, 2008

[11] International Standard ISO 7637-2, Road Vehicles - Electrical disturbances from conduction and coupling - Part 2: Electrical transient conduction along supply lines only, International Standardization Organization, 2004

5. Appendix

5.1 Attached Files

Attached to these application hints (or included in the same archive) you can find source code for an UJA1078A Core SBC example project, its software documentation and an UJA107xA power dissipation calculator. Table 17 gives an overview and a description of the attached files.

Table 17. Attached files

Attached file	Description
UJA107xA_ApplHint.c	C-file containing the main function
NXP_SBC_UJA107xA.h	General header file for UJA107xA Core SBC
NXP_SBC_UJA107xA_Functions.h	Header file defining UJA107xA specific functions
NXP_SBC_UJA107xA_Functions.c	C-file implementing UJA107xA specific functions
Application_Specific_Variables.h	Header file defining application specific variables
Application_Specific_Functions.h	Header file defining application specific functions
Application_Specific_Functions.c	C-file implementing application specific functions
uC_Specific_Functions.h	Header file defining microcontroller specific functions
uC_Specific_Functions.c	C-file containing microcontroller specific functions that must be implemented by the SW designer
Example_Software_Documentation.zip.txt	Documentation of the example project; Please remove the suffix ".txt" and extract the ZIP-archive afterwards.
UJA107xA_Power_Dissipation_Calculator_v12.xls	Excel Tool for calculating the power dissipation for the UJA107xA variants in different use cases The power dissipation calculator can only be used if macros are allowed/enabled in the Microsoft Excel Tool.

5.2 NXP UJA107xA Example Software

This example project, attached to the UJA107xA Core SBC Application Hints, reflects the software flow discussed in chapter 3 of this Applications Hints. Moreover, it shows how to program the UJA107xA in more detail. It demonstrates how to:

- Initialize the UJA107xA
- Trigger the watchdog
- Change operation modes
- Handle interrupts etc.

Please note that this software is different from the software version which is running on the UJA107xA Demonstrator hardware.

On the other hand this example project includes some application specific code which should build the frame for the UJA107xA related actions. If a user of this example code wanted to add/extend/modify the application, this must be done by changing/adding functions to example application specific functions (see 2.)). Moreover, it includes a microcontroller abstraction layer that makes this example code platform (microcontroller) independent. For that reason this example project consists of three main parts:

1. NXP UJA107xA specific functions. See File NXP_SBC_UJA107xA_Functions.c for details. It includes functions like:
   - SoftwareReset()
   - InterruptHandler()
   - SleepMode()
   - StopMode()
   - CyclicWakeUpSBC()
   - EnterNormalOperation()
   - TransceiverControl()
   - GetDeviceID()
   - LimpHomeHandling()
   - ResetAndWakeUpDetection()
   - StartupOperation()

   Theses file can be used as it is, even for different applications.

2. Example application specific functions. See File Application_Specific_Functions.c for details. It includes functions like:
   - SchedulerOnTimerOverflow()
   - WatchdogTrigger()
   - BAT_Supervisor()
   - Wake_Supervisor()
   - CheckMemoryAndApplication()
   - InitApplication()
   - InitUJA107xA()
   - SychcMicrocontrollerAndUJA107xA()
   - EnterFlashOperation()
   - AbortLowPowerHandler()
   - V1UI_InterruptService()
   - V2UI_InterruptService()

- V2polling()
- LWI1_InterruptService()
- LWI2_InterruptService()
- CI_InterruptService()
- WI1_InterruptService()
- WI2_InterruptService()
- CWI_InterruptService()
- StopCAN_TX()
- ResumeCAN_TX()
- INTN_OnFallingEdge()

This file includes application specific code and is only for demonstration purpose.

3. Microcontroller specific functions (Microcontroller abstraction layer). See File uC_Specific_Functions.c for details. It includes functions like:
   - SpiSend()
   - EnterStopMode()
   - InitMicrocontroller()
   - RamCheck()
   - FlashProgramming()
   - AbortTransmissionCAN()
   - EnableTransmissionCAN()
   - GetADCconversionResults()
   - EnableOSCandPLL()
   - ScanPort()

This file must be implemented with the microcontroller specific code.

Table 18. Cross reference table for UJA107xA specific functions

Function	Description
SoftwareReset() in NXP_SBC_UJA107xA_Functions.c	Function that performs a software reset. The software reset is triggered by writing the Software Reset (SWR) bit in the UJA107xA Watchdog and Status Register (WDSR).
InterruptHandler() in NXP_SBC_UJA107xA_Functions.c	Main interrupt entry point. This function is called by the extern interrupt. It reads the UJA107xA Interrupt Status Register (ISR). Moreover, it calls the related interrupt service routines of the pending interrupts like: - V1UI_InterruptService() related to V1 Undervoltage Interrupt (V1UI)

Function	Description
	- V2UI_InterruptService() related to V2 Undervoltage Interrupt (V2UI) - LWI1_InterruptService() related to Wake Interrupt (LWI1) - LWI2_InterruptService() related to Wake Interrupt (LWI2) - WI1_InterruptService() related to WAKE1 Interrupt (WI1) - WI2_InterruptService() related to WAKE1 Interrupt (WI2) - CI_InterruptService() related to Cyclic Interrupt (CI) - CWI_InterruptService() related to CAN Wake Interrupt (CWI) On the other hand this function also does the clean up after a failed transition to low power mode by calling AbortLowPowerHandler().

SleepMode() in NXP_SBC_UJA107xA_Functions.c	Preparation and transition to sleep mode.
	This function handles all things that are required to enter sleep mode successfully.
	- Check if Wake-Up sources are enabled by reading Interrupt Status Register (ISR). Wake-Up sources are:
	  ○ CAN Wake (STBCC)
	  ○ LIN1 Wake (STBL1C)
	  ○ LIN2 Wake (STBL2C)
	  ○ Wake (WIC1)
	  ○ Wake (WIC2)
	- Enable Wake-up sources if not done before
	- Check for pending interrupt by reading Interrupt Status Register (ISR).
	- Abort goto sleep if interrupts are pending
	- Enter sleep mode by writing Mode Control (MC) bits in the Mode Control Register (MCR).

StopMode() in NXP_SBC_UJA107xA_Functions.c	Preparation and transition to stop/sub clock mode. This function handles all things that are required to enter stop/sub clock mode successfully. - Enter Standby Mode of the UJA107xA by writing Mode Control (MC) bits in the Mode Control Register (MCR).

Function	Description

	- Check if Wake-Up sources are enabled by reading Interrupt Status Register (ISR). Wake-Up sources are:
	  ○ CAN Wake (STBCC)
	  ○ LIN1 Wake (STBL1C)
	  ○ LIN2 Wake (STBL2C)
	  ○ WAKE1 Wake (WIC1)
	  ○ WAKE2 Wake (WIC2)
	  ○ V1 Undervoltage Interrupt (V1UIE)
	- Enable Wake-up sources if not done before
	- Check for pending interrupt by reading Interrupt Status Register (ISR)
	- Abort goto stop/sub clock if interrupts are pending
	- Disable the UJA107xA watchdog timer by writing the Watchdog Mode Control (WMC) bit in the Watchdog and Status Regsiter (WDSR)
	- Disable microcontroller oscillator or switch to sub clock

CyclicWakeUpSBC() in NXP_SBC_UJA107xA_Functions.c	Preparation and transition to cyclic wake-up mode. This function handles all things that are required to enter cyclic wake-up mode successfully.
	- Enter Standby Mode of the UJA107xA by writing Mode Control (MC) bits in the Mode Control Register (MCR).
	- Check if Wake-Up sources are enabled by reading Interrupt Status Register (ISR). Wake-Up sources are:
	  ○ CAN Wake (STBCC)
	  ○ LIN1 Wake (STBL1C)
	  ○ LIN2 Wake (STBL2C)
	  ○ Wake1 (WIC1)
	  ○ Wake2 (WIC2)
	  ○ V1 Undervoltage Interrupt (V1UIE)
	  ○ Cyclic Wake (WMC)
	- Enable Wake-up sources if not done before
	- Check for pending interrupt by reading Interrupt Status Register (ISR)
	- Abort goto cyclic wake if interrupts are pending
	- Enable the UJA107xA watchdog timer by writing the Watchdog Mode Control (WMC) bit in the Watchdog

Function	Description
	and Status Register (WDSR) - Disable microcontroller oscillator or switch to sub clock mode
EnterNormalOperation() in NXP_SBC_UJA107xA_Functions.c	Transition to normal operation. This function enters Normal Mode by writing the Mode Control (MC) bits in the Mode Control Register (MCR). Afterwards the TransceiverControl() function is called.
TransceiverControl() in NXP_SBC_UJA107xA_Functions.c	Transceiver configuration in Normal Mode. Enables the CAN and LIN1/2 transceivers in Normal Mode by clearing CAN TRX Standby Control (STBCC), LIN1 TRX Standby Control (STBL1C) and LIN2 TRX Standby Control (STBL2C). Moreover, it disables the WAKE1/2 interrupts by clearing WAKE1 Interrupt Control (WIC1) and WAKE2 Interrupt Control (WIC2) to prevent an unwanted transition to Sleep Mode. This entire configuration is done by writing the Interrupt Control Register (ICR). After that the level of V2 is checked because this shows if the CAN TRX is online or not. This is done by reading the V2 Status (V2S) bit in the Watchdog and Status Register (WDSR).
GetDeviceID() in NXP_SBC_UJA107xA_Functions.c	Readout Device ID of the related family member. Determine between the different UJA107xA family members UJA1075A, UJA1076A, UJA1078A and UJA1079A by writing the CAN Transceiver Standby Control (STBCC), LIN1 Transceiver Standby Control (STBL1C) and LIN2 Transceiver Standby Control (STBL2C) bits in the Interrupt Control Register (ICR) Returns the number of family member - 0 = No Device found (error) - 1 = UJA1075A - 2 = UJA1076A - 3 = UJA1078A - 4 = UJA1079A
LimpHomeHandling() in NXP_SBC_UJA107xA_Functions.c	Limp Home Handling. This function clears the Limp Home Warning Control (LHWC) and Limp Home Control (LHC) in the Mode Control Register (MCR).
ResetAndWakeUpDetection() in NXP_SBC_UJA107xA_Functions.c	Reset and Wake-Up source detection. This function determines the reset and wake-up source by reading the Interrupt Status Register (ISR). Four different sources are detectable: - Power-on reset signaled by Power-on Status

Function	Description
	Interrupt (POSI) - Wake up out of Sleep Mode by LIN1 Wake Interrupt (LWI1), LIN2 Wake Interrupt , WAKE1 Interrupt (WI1), WAKE2 Interrupt (WI2) and CAN Wake Interrupt - V1 undervoltage or over temperature shut down - Watchdog related resets like e.g. - Extern Reset - Software Reset - Wrong Watchdog trigger - Wrong UJA107xA configuration etc. The related actions which are performed by this function depend on the detected reset source. In case of a power-on the GetDeviceID() Device ID is checked before the UJA107xA and the application is initialized. In case of Wake-up out of sleep the application is checked and the related interrupt service routine for the wake-up source is called. The V1 undervoltage reset requires a check of the application where a watchdog related reset requires a complete re-initialization of the application.
StartupOperation() in NXP_SBC_UJA107xA_Functions.c	Normal startup code of the application. This function performs the following actions, which take place after every system reset: - Reset and Wake-up source detection and related actions - Limp Home handling - Watchdog and Microcontroller time base synchronization - Transition to normal operation

5.3 Power Dissipation Calculator

The Excel document “UJA107xA_Power_Dissipation_Calculator_v12.xls” is a tool that allows estimating the power dissipation for the UJA107xA variants in different use cases. Please note, that the power dissipation calculator can only be used if macros are allowed/enabled in the Microsoft Excel Tool. For a better understanding, below you can find the formulas of the power dissipation calculator.

In general the total UJA107xA power dissipation consists of

- the quiescent power dissipation
- the power dissipation of the V1 regulator

- the power dissipation of the PNP driver (if an external PNP is used)
- the power dissipation of the LIN transceiver(s)
- the power dissipation of the CAN transceiver
- the power dissipation of the V2 regulator.

Hence the total UJA1078A power dissipation is calculated by:

$$P_{total} = P_{quiescent} + P_{V1} + P_{PNP\,Driver} + P_{LIN1} + P_{LIN2} + P_{V2} + P_{CAN}$$

Quiescent power dissipation

The SBC quiescent power dissipation contains the power needed in Normal Mode, when V2 is activated. Hence, it considers the battery current needed for an active V1 regulator (I_{BAT}@MC=00 (Standby;V1 on, V2 off)) and the additional current needed for an active V2 regulator ($\Delta I_{BAT(add)}$@V2 on; MC = 11; V2UIE = 1; IV2 = 0 mA):

$$P_{quiescent} = V_{BAT} \times (I_{BAT} + I_{BAT(add)})$$

(In case CAN is supplied out of the V1 regulator, the additional current for an active V2 regulator can be removed from the estimation.)

V1 power dissipation

The V1 power dissipation depends on the voltage at the BAT pin and the current drawn from the V1 regulator:

$$P_{V1} = (V_{BAT} - V_{V1}) \times I_{V1}$$

PNP Driver power dissipation

In case an external PNP transistor is connected to distribute the power, the PNP driver must also be considered into the power dissipation estimation:

$$P_{PNPDriver} = (V_{BAT} - V_{BE(PNP)}) \times \frac{I_{PNP}}{\beta_{PNP}}$$

Transceiver power dissipation

The power dissipation of the transceivers is in general determined by the time the transceiver is sending a dominant signal on the bus, i.e.

1) the relation between bus communication ongoing and transceiver is sending a message (node activity ratio - $p_{node\ actvity}$) and
2) the relation between dominant and recessive bits in a message (dominant ratio - $p_{dominant}$)

are important.

LIN1 power dissipation

The power dissipation of the LIN transceiver(s) is estimated by adding the dominant to the recessive power dissipation. Additionally for calculation of LIN1 power dissipation the impact of the integrated diode, which should be used for LIN1 termination, is also considered.

$$P_{LIN1} = P_{LIN1(rec)} \times \left(1 - p_{nodeactivity} \times p_{dominant}\right) + \\ + \left(P_{LIN1(dom)} + P_{DLIN(dom)}\right) \times p_{nodeactivity} \times p_{dominant}$$

$$P_{LIN1(rec)} = V_{BAT} \times \Delta I_{BAT(add)@LINx\ Active\ Mode(recessive)}$$

$$P_{LIN1(dom)} = V_{O(dom)} \times I_{pull-up} + V_{BAT} \times \Delta I_{BAT(add)@LINx\ Active\ Mode(dominant)}$$

$$V_{O(dom)} = 0{,}1111\% \times V_{BAT} \quad \text{(based on product data sheet: } 2V@V_{BAT} = 18V \rightarrow 11{,}11\%\text{)}$$

$$I_{pull-up} = \left(V_{BAT} - V_{(DLIN-BAT)} - 11{,}11\%\ V_{BAT}\right) \div R_{load}$$

$$P_{DLIN(dom)} = V_{(DLIN-BAT)} \times I_{pull-up}$$

In the *error case "LIN1 clamped to BAT"* the maximum current limitation for the LIN driver is I_{BUS_LIM} = *100mA*. Therefore the power dissipation of LIN1 in that error case can be estimated by:

$$P_{LIN1} = P_{LIN1(rec)} \times \left(1 - p_{nodeactivity} \times p_{dominant}\right) + \\ + I_{BUS\_LIM} \times V_{BAT} \times p_{nodeactivity} \times p_{dominant}$$

LIN2 power dissipation

The LIN2 transceiver needs an external termination diode. Hence, its power dissipation is estimated by:

$$P_{LIN2} = P_{LIN2(rec)} \times \left(1 - p_{nodeactivity} \times p_{dominant}\right) + \\ + P_{LIN2(dom)} \times p_{nodeactivity} \times p_{dominant}$$

$$P_{LIN2(rec)} = V_{BAT} \times \Delta I_{BAT(add)@LINx\ Active\ Mode(recessive)}$$

$$P_{LIN2(dom)} = V_{O(dom)} \times I_{pull-up} + V_{BAT} \times \Delta I_{BAT(add)@LINx\ Active\ Mode(dominant)}$$

$$V_{O(dom)} = 0{,}1111\% \times V_{BAT} \quad \text{(based on product data sheet: } 2V@V_{BAT} = 18V \rightarrow 11{,}11\%\text{)}$$

$$I_{pull-up} = \left(V_{BAT} - 1V - 11{,}11\%\ V_{BAT}\right) \div R_{load} \quad \text{(assumption: drop of external diode is 1V)}$$

In the *error case "LIN2 clamped to BAT"* the maximum current limitation for the LIN driver is I_{BUS_LIM} = *100mA*. Therefore the power dissipation of LIN2 in that error case can be estimated by:

$$P_{LIN2} = P_{LIN2(rec)} \times \left(1 - p_{nodeactivity} \times p_{dominant}\right) + \\ + I_{BUS\_LIM} \times V_{BAT} \times p_{nodeactivity} \times p_{dominant}$$

V2 and CAN power dissipation

As the CAN transceiver is usually supplied by the V2 regulator, this use case is considered in the power dissipation calculator. In case the CAN transceiver is supplied by the V1 regulator the following V2 power dissipation must be added to the V1 power dissipation and the V2 power dissipation is 0W. As the CAN transceiver requires maximum 100mA (short circuit peak current of the transceiver), at least 20mA are available for other ICs. This option is also considered in the power dissipation estimation formula:

$$P_{V2} = P_{V2(CANrec)} + P_{V2(CANdom)} \times p_{nodeactivity} \times p_{dominant} + P_{V2(add)}$$

$$P_{V2(CANrec)} = (V_{BAT} - 5V) \times \Delta I_{BAT(add)@CAN\ Active\ Mode(recessive)}$$

$$P_{V2(CANrdom)} = (V_{BAT} - 5V) \times I_{load(dom)}$$

$$I_{load(dom)} = (5V - V_{V2-CANH} - V_{CANL-GND}) \div R_{CANH-CANL}$$

With the assumption that $V_{O(dom)}$(CANH) = 3.5V and $V_{O(dom)}$(CANL) = 1.5V this formula can be reduced to:

$$I_{load(dom)} = 2V \div R_{CANH-CANL}$$

$$P_{V2(add)} = (V_{BAT} - 5V) \times I_{V2(add)}$$

During CAN communication further power is dissipated in the CAN transmitter:

$$P_{CAN} = (V_{V2-CANH} + V_{CANL-GND}) \times I_{load(dom)} \times p_{nodeactivity\ ratio} \times p_{dominantratio}$$

With the assumption that $V_{O(dom)}$(CANH) = 3.5V and $V_{O(dom)}$(CANL) = 1.5V this formula can be reduced to:

$$P_{CAN} = 3V \times I_{load(dom)} \times p_{nodeactivity\ ratio} \times p_{dominantratio}$$

In the *error case "CANH clamped to GND"* the maximum CAN dominant output current is $I_{O(dom)} = 100mA$. Therefore the power dissipation for V2 and CAN transmitter can be estimated by:

$$P_{V2} = P_{V2(CANrec)} + (V_{BAT} - 5V) \times I_{O(dom)} \times p_{nodeactivity} \times p_{dominant} + P_{V2(add)}$$

$$P_{CAN} = 5V \times I_{O(dom)} \times p_{nodeactivity\ ratio} \times p_{dominantratio}$$

6. Legal information

6.1 Definitions

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

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

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 accepts 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.

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 national authorities.

6.3 Trademarks

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

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

Date of release: 17 March 2016

Document identifier: DOC-151967