Wake-up implementation for TLE9255W and TLT9255WLC

High speed CAN transceiver with Partial Networking

About this document

Scope and purpose

This document provides additional information for the TLE9255WLC, TLE9255WSK and TLT9255WLC. This document relates to the following datasheets:

- TLE9255W datasheet Rev. 1.03 (TLE9255WLC, TLE9255WSK).

- TLT9255WLC datasheet Rev. 1.00.

The latest datasheets can be found here:

https://www.infineon.com/cms/en/product/transceivers/automotive-transceiver/automotive-can-transceivers/tle9255wlc/

https://www.infineon.com/cms/en/product/transceivers/automotive-transceiver/automotive-can-transceivers/tle9255wsk/

https://www.infineon.com/cms/en/product/transceivers/automotive-transceiver/automotive-can-transceivers/tlt9255wlc/

It is strongly recommended to follow the guidelines of this document when using one of the above mentioned devices.

Intended audience

This document is intended for engineers who develop applications using TLE9255W and TLT9255WLC.

Table of contents

1 Details: the possibility of an unexpected state

The TLE9255W and TLT9255WLC are high speed CAN FD transceivers compliant to ISO 11898-2:2016 with Partial Networking.

During Sleep mode, if a SPI-driven wake-up command is sent by the host at the same time as a WUF is detected, the transceiver's SPI registers may signal the state "normal" whereas the state machine has transitioned to standby mode. In standby, CAN Bus transmission is disabled, so if the controller triggers a CAN message the message is not transferred to the Bus. Figure 1 illustrates the situation: the dotted lines show the SPI command to switch to normal mode or the detected WUF which leads to a state change to standby mode.

This document explains the background of this problem – how and when this can happen – and explains also how to solve it.

Figure 1   **State machine according to the referenced data sheet. Dotted lines signal the SPI driven state change and the WUF driven state change.**

1.1 Explanation of the possibility of an unexpected state

When the device is in Selective Wake Sub-Mode a CAN frame is able to wake up the ECU. Details how to configure this mode are explained in the data sheet in chapter 5.4.2 as well as in chapter 7.

The state change from Selective Wake Sub-Mode to normal mode is issued via SPI command. If a WUF is detected within the same 200 ns clock cycle where the SPI frame is finished (CSN high), the device does not change to normal mode as requested via SPI but changes to standby mode as requested by the WUF. This behavior is correct.

However, in this case, the MODE_CTRL register does not indicate the correct mode of standby. Instead the mode is indicated as Normal, as issued by the SPI command but not successfully finished. If a microcontroller reads the status information from the SPI register, MODE_CTRL defines the state as "normal mode", but the transceiver's internal state machine is in standby mode. Figure 2 shows a timing diagram illustrating the situation.

Even though the described behavior is very unlikely to happen, to avoid such a race condition, two possible workarounds are described in chapter 2.
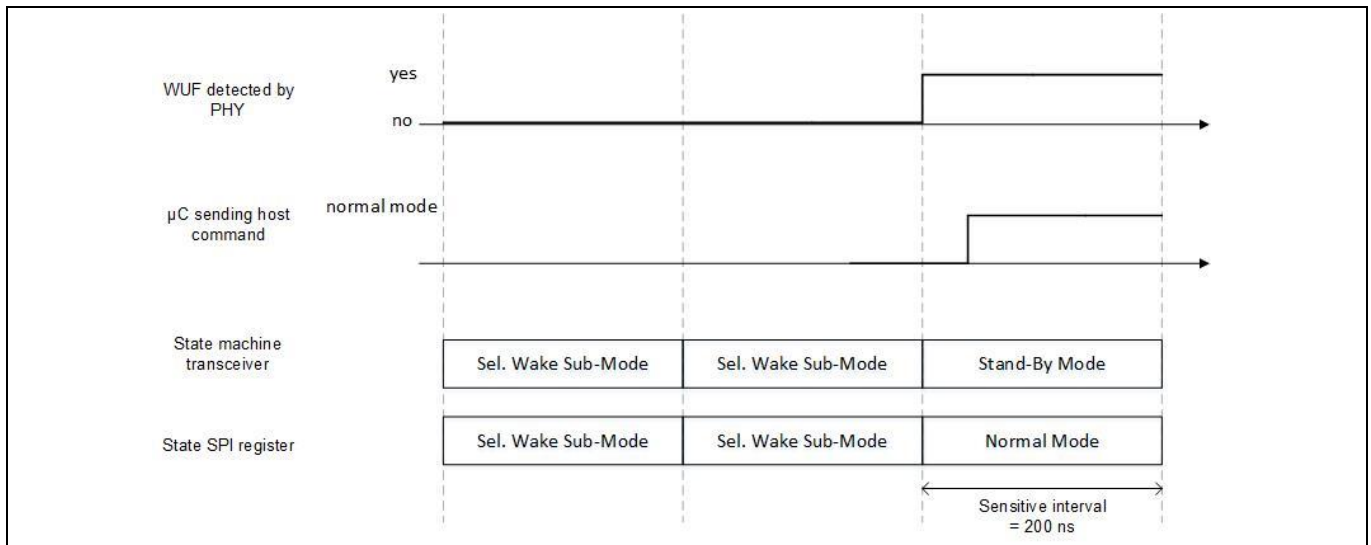

Figure 2   **WUF detected in same interval as SPI command, race condition happens.**

Wake-up implementation for TLE9255W and TLT9255WLC
High speed CAN transceiver with Partial Networking
Table of contents

2 Proposed workaround

Two different solutions are proposed to avoid the above described race condition.

2.1 Solution 1: check register >Wake Status<

In case the state change to normal mode shall be initiated by a SPI command, one solution to avoid the negative consequence of a race condition would be to check the status of the register *Wake Status* WAKE_STAT (1B$_H$). This register includes the information whether a WUF was detected or not (see Figure 3).

If the value of the field WUF is equal to 1$_B$, the above described scenario may have happened. It is therefore recommended to send the SPI command for switching to normal mode a 2nd time. If the device is already in normal mode, another mode switching command will not effect the behavior of the device. If the device is not yet in normal mode but in standby mode, issuing the command again triggers a state change to normal mode. Figure 4 illustrates this process.
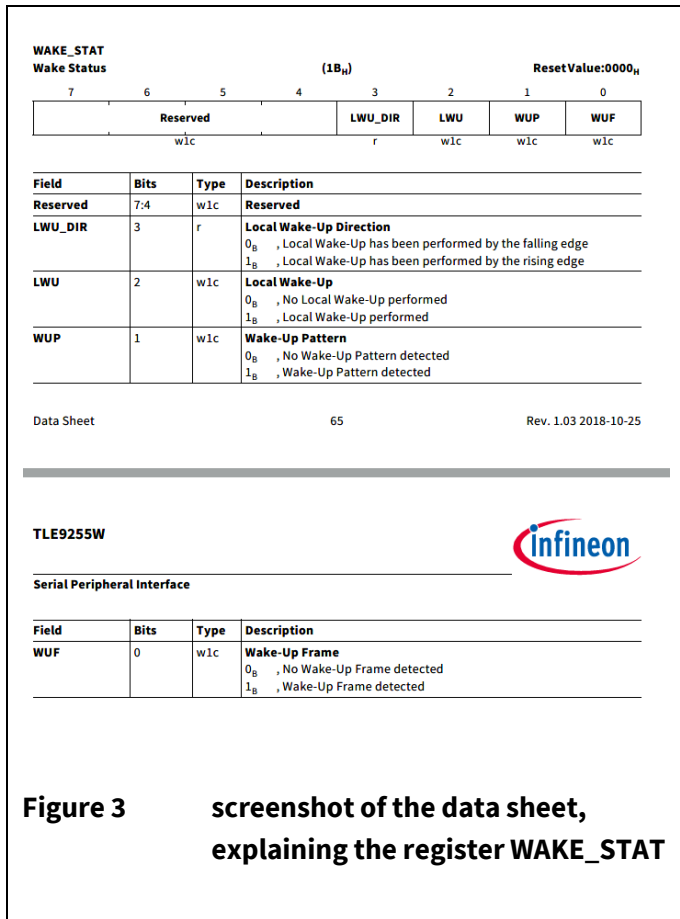
WAKE_STAT							
Wake Status			(1B$_H$)		Reset Value:0000$_H$		
7	6	5	4	3	2	1	0
Reserved				LWU_DIR	LWU	WUP	WUF
	w1c			r	w1c	w1c	w1c

Field	Bits	Type	Description
Reserved	7:4	w1c	Reserved
LWU_DIR	3	r	**Local Wake-Up Direction** 0$_B$, Local Wake-Up has been performed by the falling edge 1$_B$, Local Wake-Up has been performed by the rising edge
LWU	2	w1c	**Local Wake-Up** 0$_B$, No Local Wake-Up performed 1$_B$, Local Wake-Up performed
WUP	1	w1c	**Wake-Up Pattern** 0$_B$, No Wake-Up Pattern detected 1$_B$, Wake-Up Pattern detected

Data Sheet 65 Rev. 1.03 2018-10-25

TLE9255W

Serial Peripheral Interface

Field	Bits	Type	Description
WUF	0	w1c	**Wake-Up Frame** 0$_B$, No Wake-Up Frame detected 1$_B$, Wake-Up Frame detected

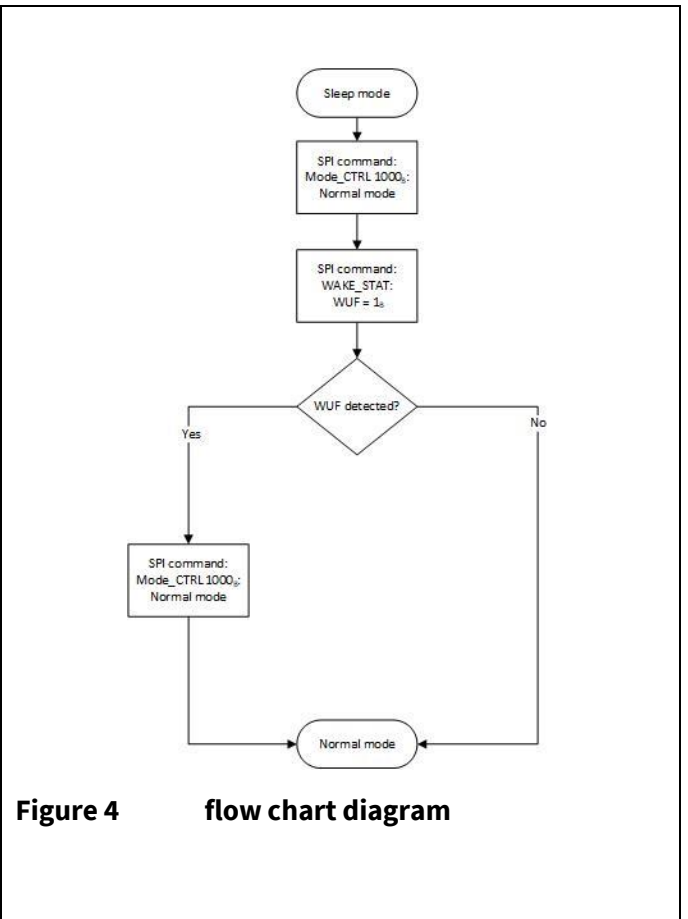**Figure 3** screenshot of the data sheet, explaining the register WAKE_STAT

Figure 4 flow chart diagram

2.2 Solution 2: sending the SPI command to normal mode 2 times

Another option to avoid the race condition is to simply send the SPI command for switching to normal mode two times. If the device is not yet in normal mode but in standby mode, issuing the command again triggers a state change to normal mode. If the device is already in normal mode, another mode switching command will not effect the behavior of the device. Please be aware of calculating t$_{Mode\_Change}$ (max. 20 µs) accordingly when implementing this solution.

Revision history

Document version	Date of release	Description of changes
1.1	2020-06-23	Errata sheet updated, figures added.
1.0	2020-06-04	Errata sheet created.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.