Possible Loss Of Wakeup Message

Version 1.0
2008-02-26
Application Note AN-ISC-8-1074

Author(s) Ralf Fritz

Restrictions Customer confidential - GM only

Abstract This application note describes the possibility of loosing a wake-up request in a GMLAN Single Wire environment and the possibility to minimize the possibility of occurrence. For GM customer only.

Table of Contents

1.0 Overview

GMLAN uses special messages, so called High Voltage Wake-Up Messages (HVWU), to wake-up sleeping modules on the Single Wire CAN bus. Under certain conditions it is possible that these messages are discarded and do not cause a wake-up of the ECU like expected. This document describes these possible situations and methods of reducing the time period where a HVWU message can be lost.

Note: This document applies to the Single-Wire CAN bus only, because of its special physical implementation. The wake-up behavior on a non Single-Wire CAN bus will differ since every receive message will cause a CAN cell wake-up if the CAN cell supports sleep wake-up.

Note: The occurrence of this behavior depends on the hardware platform, the compiler, hardware layout and many more parameters. Because of this, this application note cannot specify specific time durations.

Note: **Any code samples are only examples of implementation proposals. Because these functions are meant for demonstration purposes only, Vector's liability shall be expressly excluded in cases of ordinary negligence to the extent admissible by law or statute.**

2.0 Initialization Of CAN controller

The GMLAN handler will initialize the CAN controller at different times during runtime. The CAN controller is initialized in the following situations:

- During startup of the handler
- Switching baud-rate during flash programming
- After a Bus-Off event
- Before the handler tries to enter the sleep mode
- In case the High Voltage Wake-Up message cannot be sent onto CAN bus

If the CAN cell is initialized, all currently received and not handled receive-messages will be discarded. This means that also pending HVWU messages will be discarded and can cause the module not to switch back to Communication Enable or Communication Active state as expected.

The chapters below include a more detailed description of these situations.

2.1 During Startup

Vectors GMLAN handler can be configured to initialize the CAN controller during the call of `IlInitPowerOn()`. During this time it is not likely to lose any HVWU message, because this function is only called if the system performs a cold start. Please see [1] and [2] for further information when this function needs to be called by the application.

After the call of `IlInitPowerOn()`, the handler will enter at least the Communication Enable state for 8 seconds, this means bus activity can be detected and a lost HVWU frame will not cause the module not to participate in the CAN communication.

2.2 Switching Baud Rate

To be able to switch the current baud rate to a higher or lower one, the CAN controller needs to be re-initialized with the different baud rate settings. During this period, the handler is in the Normal Communication Halted state. This means that the module is already awake during this time and will stay awake for at least 4 more seconds. This means that if a HVWU frame is lost at this point in time, the handler will still be able to participate in an ongoing CAN communication.

2.3 Bus-Off Event

To start the CAN cell bus-off recovery sequence and to remove any pending transmit request from the CAN controller, the GMLAN handler is calling two CAN Driver interface functions. These functions are `CanResetBusOffStart()` and `CanResetBusOffEnd()`. One of these functions is usually mapped to a function which will re-initialize the CAN cell. In such situations, the handler shall ignore any receive messages, which also includes HVWU messages.

Note: The implementation of the functions `CanResetBusOffStart()` and `CanResetBusOffEnd()` are depending on the hardware. Please check the related hardware manual [3] for more information.

2.4 Before Sleep Mode

Vectors GMLAN handler will call `CanResetBusSleep()` before the CAN controller is put into sleep mode to remove any pending transmit request from the CAN hardware.

Note:	The implementation of the function CanResetBusSleep() is depending on the hardware. Please check the related hardware manual [3] for more information.

This function is usually mapped to a function which will initialize the CAN controller. The call of this function happens when the remainder of the Communication Enable timer reaches a value of 100 ms + 2 times the task cycle of the network management task (`IlNwmTask()`) and the application accepts the sleep request (`ApplNwmSleepConfirmation` returns `NmSleepOk`).

Note:	100 ms is the default provided by the generation tool. This value can be calibrated later to a different value. Please check your usage.

If now the delay between the related I-VNMF to the discarded HVWU message is longer then the time set-up, the transceivers will be switched into sleep mode before the I-VNMF is sent to the CAN bus. This will prevent the CAN controller from receiving the message and the system will stay in sleep mode. This behavior cannot be changed by a software or hardware change.

3.0 Switching Transceiver to Sleep mode

On a Single Wire bus it is quite normal, that a single ECU will enter the CAN sleep mode and other modules continue to communicate on the same physical CAN bus. To be able to do this, the transceivers of these modules are switched into sleep mode before the CAN cell is put into sleep mode. This will prevent any normal message communication from being passed to the controller and allows it to enter the CAN sleep mode. Vectors GMLAN handler does not check, if there is an ongoing CAN message reception while it requests the application to switch the transceiver into sleep mode by calling the function `ApplTrcvrSleepMode()`. If a normal receive message is interrupted, this causes no functional issue, since the module already decided that normal message reception must be ignored.

If during this time a HVWU message is received, the message is most likely lost. This is because only parts of the physical CAN message are received by the CAN controller. The transceiver will act like a filter after it is put into sleep mode and will not pass all bits of the message to the CAN controller (Please check your transceiver manual for the exact behavior of your transceiver). Because the message is not received by the CAN cell, the system will not switch back into the Communication Enable or Active state as expected.

Depending on the used hardware there are maybe some possibilities to minimize the window in which this issue can happen. Below two current know methods.

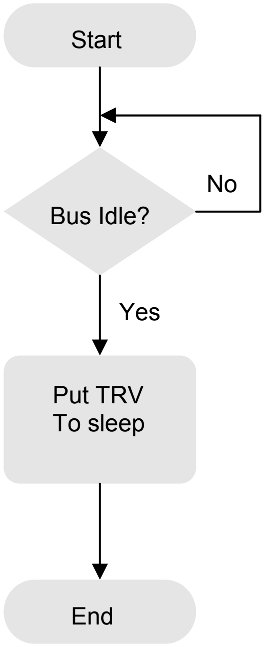
3.1 Check For Messages

The goal of this implementation is to minimize the possibility that the HVWU message is cut off by switching the transceiver into sleep mode. This requires the CAN hardware to be able to provide the information that there is currently ongoing message reception. Vectors GMLAN handler does not provide this information!

Note:	Not all hardware platforms provide the needed hardware feature for this implementation. Please check your device manual for further information.

The idea is to check for CAN bus-idle before the transceiver is switched into sleep mode. This would reduce the window of losing a HVWU message from the message length of this message to the time it takes to switch the transceiver and CAN into sleep mode after an idle CAN bus was detected.

The flow chart below shows a possible implementation strategy.



Note:	Waiting for bus idle can take a longer period of time. Please make sure that the loop has a timeout criteria which will allow the software to continue in case there will be no bus-idle detected after a certain time!

The code which implements this flow chart needs to be placed into the application callback function `ApplTrcvrSleepMode()` of the Single Wire CAN. This function is called by Vectors GMLAN handler to switch the transceiver into sleep mode. After this call, the handler will try and put the CAN cell into sleep mode.

3.2 Use Transceiver Wake Pin

If the hardware allows, it is preferable to use the wake-up notification pin of the transceiver. For most of the transceivers they provide a dedicated hardware output, which is set high, if the transceiver detects a wakeup frame on the CAN bus. If the controller detects that this pin was set, the software should call the GMLAN handler API function `NmCanWakeUp()` to notify the GMLAN handler about this event.

Note:	Not all hardware provides the needed feature for this implementation. Please check your device / transceiver manual for further information.

4.0 Switching CAN cell into sleep mode

After the transceiver is put into sleep mode the CAN cell will be put into sleep mode as well by the GMLAN handler. Depending on the hardware and the related software implementation of the GMLAN handler there is a chance that a pending HVWU message will be lost.

This can happen if the reception of a HVWU message is finished during or delayed until the time when the GMLAN handler switches the transceiver and the CAN cell into sleep mode (The GMLAN handler will disable interrupts while it is putting the CAN bus into sleep mode). In such a situation the handler will call the CAN Driver API function `CanSleep()` while a not handled receive message is pending in the CAN cell. Depending on the used CAN cell and the implementation of the handler function `CanSleep()`, this message reception is discarded or suppressed and will not lead to a wake-up. Please check the related CAN Driver manual [3] for more information about the behavior of `CanSleep()`.

5.0 Polling versus Interrupt

The GMLAN handler allows receiving message either on interrupt or polling basis. It is more likely, that a HVWU message will be lost, if the handler is configured to receive this message in polling mode. This is because usually a receive notification interrupt can happen right after the reception of the physical message and is not delayed until the next call of the receive polling task. Interrupt handling of the HVWU frame does not guarantee that the message will be received and processed all the time, since the GMLAN handler and the application potentially disable the CAN interrupts during certain periods of times for example to guarantee data consistency which will also cause a delay of the receive message handling.

6.0 Additional Resources

[1] GMLAN Technical Reference documentation.
[2] IL Technical Reference documentation.
[3] CAN Driver Technical Reference documentation of the used hardware platform.

7.0 Contacts

Vector Informatik GmbH
Ingersheimer Straße 24
70499 Stuttgart
Germany
Tel.: +49 711-80670-0
Fax: +49 711-80670-111
Email: info@vector-informatik.de

Vector CANtech, Inc.
39500 Orchard Hill Pl., Ste 550
Novi, MI 48375
USA
Tel: +1-248-449-9290
Fax: +1-248-449-9704
Email: info@vector-cantech.com

VecScan AB
Lindholmspiren 5
402 78 Göteborg
Sweden
Tel: +46 (0)31 764 76 00
Fax: +46 (0)31 764 76 19
Email: info@vecscan.com

Vector France SAS
168 Boulevard Camélinat
92240 Malakoff
France
Tel: +33 (0)1 42 31 40 00
Fax: +33 (0)1 42 31 40 09
Email: information@vector-france.fr

Vector Japan Co. Ltd.
Seafort Square Center Bld. 18F
2-3-12, Higashi-shinagawa,
Shinagawa-ku
J-140-0002 Tokyo
Tel.: +81 3 5769 6970
Fax: +81 3 5769 6975
Email: info@vector-japan.co.jp

Vector Korea IT Inc.
Daerung Post Tower III, 508
Guro-gu, Guro-dong, 182-4
Seoul, 152-790
Republic of Korea
Tel.: +82-2-2028-0600
Fax: +82-2-2028-0604
Email: info@vector-korea.com