Efficient initial wake-up methods when using Partial Networking In-Vehicle network

1st Hyun Kyung Roh
IFKOR TEC ATV VAC PWR
Infineon Technologies Korea
Seoul, South Korea
HyunKyung.Roh@infineon.com

2nd Sung Hoon Kim
IFKOR TEC ATV VAC MC
Infineon Technologies Korea
Seoul, South Korea
Wayne.Kim@infineon.com

3rd Yeon Yong Shin
IFKOR TEC ATV VAC PWR
Infineon Technologies Korea
Seoul, South Korea
Edwin.Shin@infineon.com

***Abstract*—Reducing CO2 emissions for vehicle becomes an important in Automotive industry. As a global trend, most countries are restricting CO2 emissions by law. Not all ECUs are constantly needed when the vehicle is running. A lot of ECUs consume the power even if they are not needed. Partial networking allows deactivate the ECUs which are not currently needed. And it can enable the ECUs when it is needed. It leads to reduced CO2 emissions and reduced energy consumption in vehicle system. Therefore, most OEMs apply this CAN partial networking function for their In-vehicle network. Using general CAN communication for the initial wake-up sequence, if the Master-ECU does not receive an ACK, it automatically sends CAN wake-up messages until the error counter reaches the max value. In case of all the Sub-ECUs are in the Sleep mode with Partial networking, invalid wake-up message leads to the max error counter and even it has higher possibility to entering the bus off state. In this paper, it suggests the methods that optimize the wake-up sequence through manage the errors and reduce the current consumption when all Sub-ECUs wake up from Sleep mode for the initialization**

Keywords—Networking system, Partial networking, CAN transceiver, In-Vehicle network

I. Introduction

Global climate change is one of the worldwide major challenges of these days. Among the causes of change, CO2 emissions also occupies a large proportion. Road travel accounts for three-quarters of transport emissions. Most of this comes from passenger vehicles. Cars and buses which contribute 45.1%. The other 39.4% comes from trucks carrying freight. [1] Therefore, reducing CO2 emissions in the automotive industry has become a very important point.

Partial networking is one of the efficient ways to save energy and reduce CO2 emissions. It optimizes the operating ECUs. Implementing this function allows to deactivate ECUs that are not in use and activate them only when needed. Specifications related to Partial networking are specified in ISO11898-2:2016. [2]

The partial networking works with Wake-up Frame which consists of Classical CAN. Each node is given a specific ID and data and when it enters sleep mode, that is low power mode, it can be woken up only through this matching ID and data.

The general process of partial networking has several steps. First, have to enable the Partial networking function in CAN transceiver. And set the Each ECU with specific ID and Data. Also has to mask the ID to be decoded. User should always check for errors during the setup for correct operation. After confirming that there are no errors, send CAN to sleep mode. When this Sub-ECU needs to be used and woken up, Master-ECU has to send the right corresponding ID and data to wake it. If all Sub-ECUs are in sleep mode, the Master-ECU cannot receive an ACK at the first. To handling the There is no specific process for handling this error now. Therefore, this paper will discuss the initial wake-up sequence and how to handle this error.

The general method to wake-up Sub-ECU from sleep mode is using general CAN communication method. It provides a means for automatic retransmission of frames that have failed arbitration, are not acknowledged, or have been interrupted by errors in transmission. In this case, if an ACK error continues to occur due to a wrong wake up message, the error count will go up to the maximum value. In addition, if another error occurs during this transition time, it has higher possibility to move to Bus off state.

The first proposed method is using the MCU DAR (Disabled Automatic Retransmission) function. After disabling automatic retransmission function, users can manage CAN messages during error conditions and error counts.

The second proposed method is using the internal Loop-Back Mode. In this case, ACK is sent from another CAN module inside the Master-ECU. Therefore, ACK error does not occur.

In this paper, all of the following topics will be discussed. Firstly, A simple block diagram of the partial networking architecture and its process is described. Secondly, the paper discusses the current solution that is using general CAN communication method. Thirdly, the paper presents the proposed of the wake-up process by using two different methods to manage error during the wake up. In the end, the paper compares the all of the current process and proposed methods. And it shows the proposed methods can manage the error and reduce current consumption which leads the initial wake-up more efficiently.

II. Partial Networking Process

The block diagram of the simple partial networking architecture is shown in Figure 1.

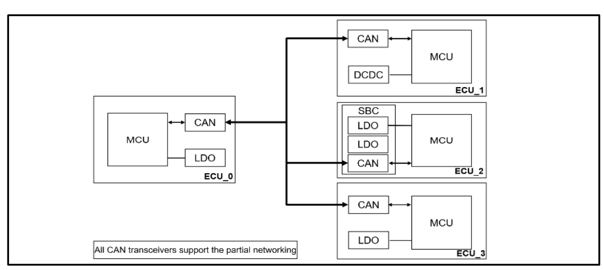

Fig. 1. Partial networking architecture

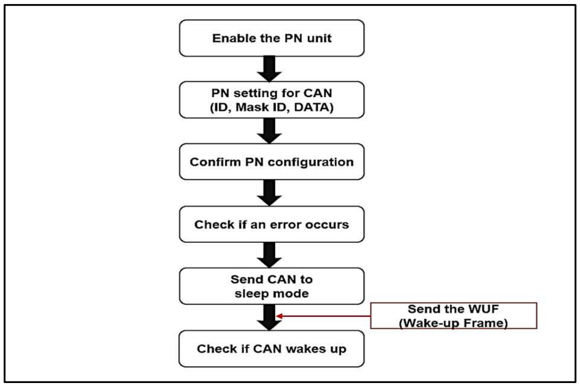

Fig. 2. Partial networking process

To enable the partial networking feature, it requires the process described in Figure 2. For the Sub-ECUs to wake up correctly, it must send a valid wake-up frame. [3]

The block diagram of the CAN error tracking is shown in Figure 3. TEC is Transmit error counter and REC is Receive error counter. Error Active state is normal state. The REC and TEC counters exceed 127 then the state moves to the Error Passive. If the TEC error counter exceeds 255, the state transitions to Bus off. [4]

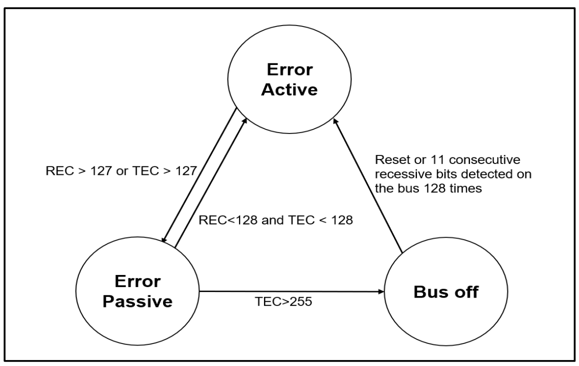

Fig. 3. CAN error tracking

III. General CAN Communication Method

According to CAN specification (ISO 11898-1, 6.3.3 Recovery management), MCU provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission.

Sub-ECUs are waiting for the wake-up message in the sleep mode after the partial networking configuration. For initial wake-up, if the CAN message is valid WUF, the ACK error will occur when sending the first message because none of the Sub-ECUs can send an ACK because all of them are in sleep mode. And the second message wakes up the specific Sub-ECU. After it wakes up, it can communicate normally. In case of the CAN message is invalid WUF, the Master-ECU will continue to send message until it receives an ACK which is shown in Figure 4. However, no Sub-ECUs wake up and send no ACK because it cannot wake up with invalid wake-up message. Therefore, when only an ACK error occurs, a waring error is raised at TEC 0x60 and goes up to 0x80. If other errors such as bit error occur in the process, it has potential to go Bus off state. CAN Bus off is a critical error that results in loss of communication. It occurs when the error count increments the most, rather than any other error state

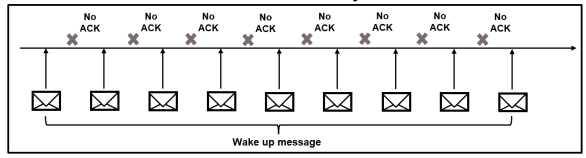

Fig. 4. General CAN communication method wake-up process with invalid WUF

IV. Proposed Methods for Initial Wake-up Process

A. *Microcontroller Disabled Automatic Retransmission function*

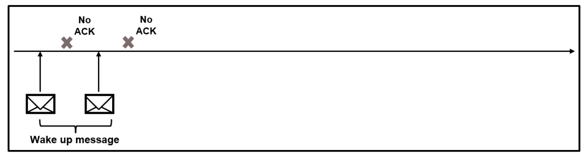

Fig. 5. DAR wake-up process with invalid WUF

DAR is Disabled Automatic Retransmission. This means user can manage the sending CAN message frame.

Same as general CAN communication method, the Master-ECU sends the WUF for the Initial wake-up. If using the DAR feature, user can read the error type after each frame is sent. Therefore, after sending two wake-up messages, if the error types are ACK errors and the Sub-ECU wakes up, it can be considered as two messages were for wake-up and they were valid.

User also can control the time interval and number of messages with DAR. It allows user to manage the error counter which means it has less possibilities to get the warning message and move to Bus off.

In addition, it can reduce the current consumption compared to the general CAN communication case, due to it manages the number of sending messages.

B. *CAN internal Loop-Back Mode*

The Microcontroller CAN module provides a Module internal Loop-Back Mode which is shown in Figure 6. The loop-back feature consists of an internal CAN bus (inside the Microcontroller CAN module) and a bus select switch for each CAN node. With the switch, each CAN node can be connected either to the internal CAN bus (Internal Loop-Back Mode activated) or the external CAN bus, respectively to transmit and receive pins (normal operation). All CAN nodes that are in Loop-Back Mode may communicate together via the internal CAN bus without affecting the normal operation of the other CAN nodes that are not in Loop-Back Mode. [5]

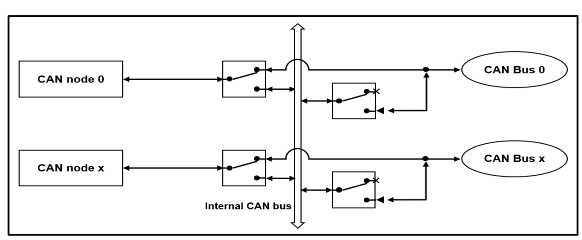

Fig. 6. Module internal Loop-Back Mode

Activate the loop-back module in the Master-ECU to connect the CAN node 0 and CAN node x internally. After partial networking configuration and all Sub-ECUs are in the sleep mode, the Initial wake-up sequence starts when detect the WUF. In this method, since CAN node x sends ACK instead of Sub-ECUs, no ACK error occurs during the initial wake-up. In addition, it can decode whether it is a valid WUF. Therefore, if the sending message is an invalid WUF, it can be retransmitted with modified WUF.

After confirming that the Sub-ECU is awake, this function can be disabled to detect ACK errors that may appear during the normal operation.

In this case, implementation is difficult if the number of nodes is not sufficient.

V. TEST RESULT AND COMPARE

The tests were conducted through Microcontroller (Aurix) and System Basis Chip. System Basis Chip is system power supply which includes the linear regulators and CAN. The logging environment is T32. One WUF takes about 230us to transmit as shown in Figure 7.

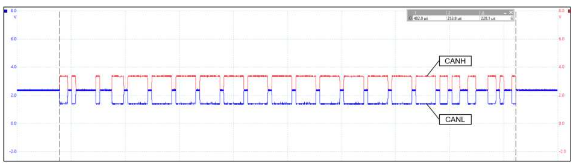

Fig. 7. WUF one frame time

Sub-ECU's CAN which is inside in SBC wake-up sequence is shown in Figure 8. Reset goes up 10ms after main power VCC1 reaches the threshold voltage. [6]

After that initialization sequence which configured with Serial peripheral interface which can be observed with CS(Chip Select) is needed to prepare the CAN communication. Finally, CAN Bus (CANH and CANL) can be activated. It takes about the 13ms for first CAN message activation after wake-up sequence starts.

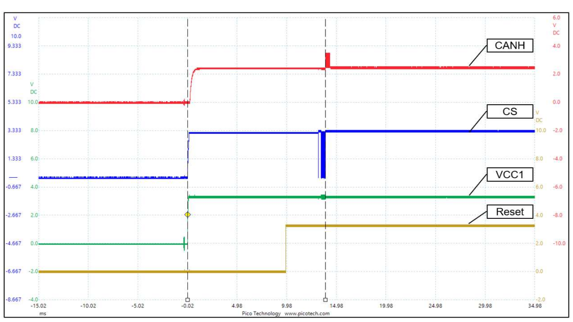

Fig. 8. CAN Bus sending time

At least two messages are required for the initial wake up the when all Sub-ECUs are in the sleep mode. The first message is to activate the CAN bus and the second message is to wake the Sub-ECUs from sleep mode.

General CAN communication method wake-up process is shown in Figure 9. and Figure 10. Using this method, Master-ECU send the messages automatically until Sub-ECU responds by sending ACK.

In case of valid message, it wakes up the Sub-ECU after two messages, which is the minimum number required for initial wake up. Therefore, the main supply VCC1 raised up which is shown in Figure 9. However, Sub-ECU can respond after 13ms due to the initialization sequence. As mentioned, one frame takes about 230us. Therefore, almost 57 WUF frames are sent from the Master-ECU automatically until Sub-ECU responds.

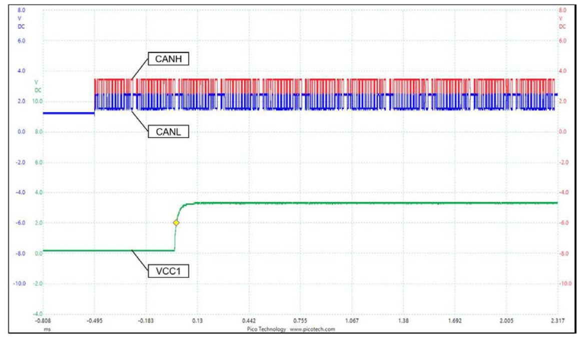

Fig. 9. General CAN communication method with valid WUF

In case of an invalid message, Sub-ECU will remain in the sleep mode. In sleep mode, the main power supply VCC1 is not powered up whish is shown in Figure 10. Therefore, Sub-ECU cannot respond. It causes more frames to be transmitted after 57 WUF frames.

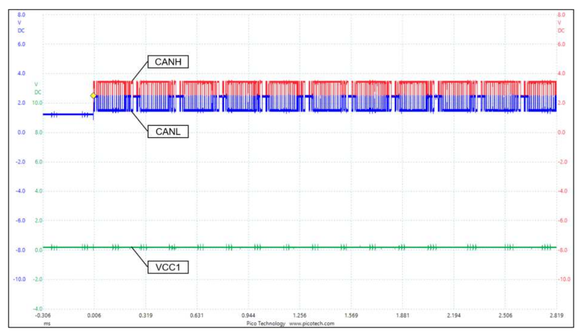

Fig. 10. General CAN communication method with invalid WUF

Error log of the general CAN communication method is shown in Figure 11. Both valid and invalid WUF cases, if there are no other errors during transmission, an error warning is raised and the TEC is raised to 0x80. And the error type would be ACK error. As mentioned, if any errors other than ACK errors occur on the bus during this process, there is a high possibility that move to the bus off state.

LBCK	Disabled		
EFBI	Disabled	PXHD	No
TEST	Disabled	DAR	No
CSA	Not acknowledged	ASM	Normal
NTSEG1	0E	NTSEG2	03
ETOC	Disabled		
REC	00	TEC	80
RFDF	Not received	RBRS	Not set
BO	Not in Bus_Off	EW	Warning
LEC	Ack error		

Fig. 11. General CAN communication method Error log

DAR wake-up process is shown in Figure 12. and Figure 13. As mentioned, using DAR function allows manage the sending message.

Therefore, Master-ECU can manage the number of messages it sends. After Master-ECU sends two messages which is the minimum number required for wake up, Sub-ECU wakes up if the messages are valid. Whether this waking was performed properly can be checked by whether the main voltage VCC1 has risen which is shown in Figure 14. Sub-ECUs can respond after 13ms such as general CAN communication method due to the initialization sequence.

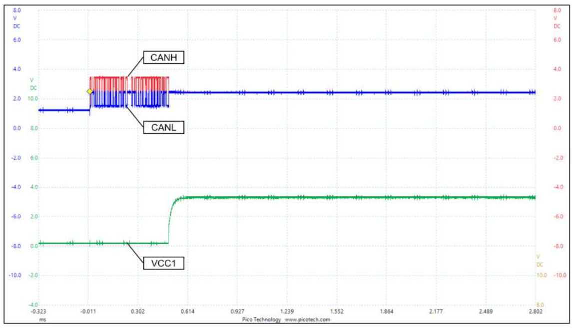

Fig. 12. DAR wake-up process with valid WUF

VCC1 is not rise up with invalid message and remains in the sleep mode as shown in Figure 13. User can manage and send the message again if the Sub-ECU does not wake up after sending this two WUF frames.

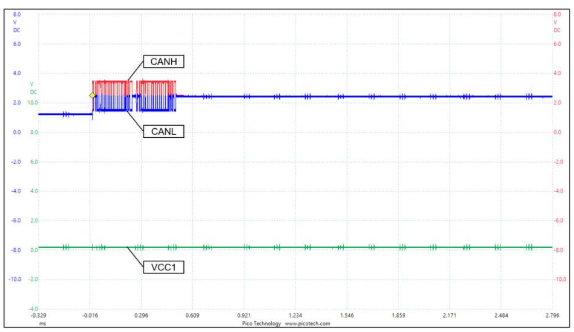

Fig. 13. DAR wake-up process with invalid WUF

Error log of the DAR wake-up process is shown in Figure 14. Both valid and invalid WUF cases with sending only two messages, if there are no other errors during transmission, an error warning is raised and the TEC is raised to 0x10. And the error type would be ACK error. If the message is invalid so user send the more than two messages the TEC will be higher.

LBCK	Disabled		
EFBI	Disabled	PXHD	No
TEST	Disabled	DAR	Yes
CSA	Not acknowledged	ASM	Normal
NTSEG1	0E	NTSEG2	03
ETOC	Disabled		
REC	00	TEC	10
RFDF	Not received	RBRS	Not set
BO	Not in Bus_Off	EW	No warning
LEC	Ack error		

Fig. 14. DAR wake-up process Error log

CAN module internal connection wake-up process with valid WUF is shown in Figure 15 and Figure 16.

In case of valid message, Sub-ECU wakes up after two frames. Sub-ECU's main power VCC1 is rise up as shown in Figure15. The internal CAN module sends the ACK instead of Sub-ECU during the initial wake-up. Therefore, no additional WUF are sent from the Master-ECU automatically. Due to the initialization sequence, Sub-ECU can respond after 13ms such as general CAN communication method.

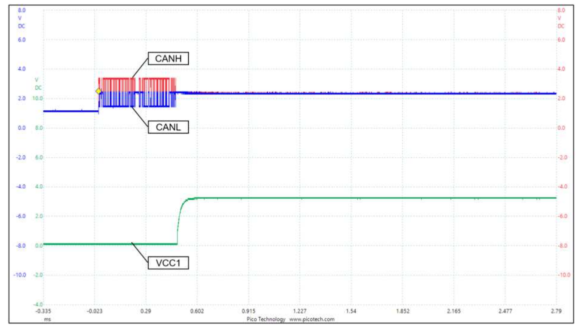

Fig. 15. CAN internal Loop-Back Mode with valid WUF

Sub-ECU will be in the sleep mode, if the message is invalid. The main supply VCC1 will maintain 0V which is shown in Figure 16. In this case, the internal CAN module transmitted the ACK instead of Sub-ECU during the initial wake-up. Therefore, Master-ECU does not send the additional WUF automatically.

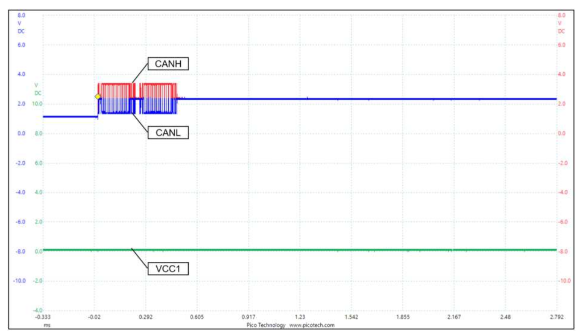

Fig. 16. CAN internal Loop-Back Mode with invalid WUF

Error log of the CAN module internal connection process is shown in Figure 11. As mentioned, the internal CAN module can send the ACK instead of Sub-ECU and monitor the transmitted messages. Therefore, the TEC is not increment due to ACK errors. And there is no ACK error. User has to disable the internal connection after the initial wake up. It cannot detect the ACK error if still connected in normal CAN communication.

LBCK	Disabled		
EFBI	Disabled	PXHD	No
TEST	Disabled	DAR	No
CSA	Not acknowledged	ASM	Normal
NTSEG1	0E	NTSEG2	03
ETOC	Disabled		
REC	00	TEC	00
RFDF	Not received	RBRS	Not set
BO	Not in Bus_Off	EW	No warning
LEC	No error		

Fig. 17. CAN internal Loop-Back Mode Error log

The CAN module internal connection wake-up process allows user to monitor the transmitted message which is shown in Figure 18. The user can recognize whether the WUF is valid or invalid by comparing the transmitted WUF with the monitored ID and DATA.

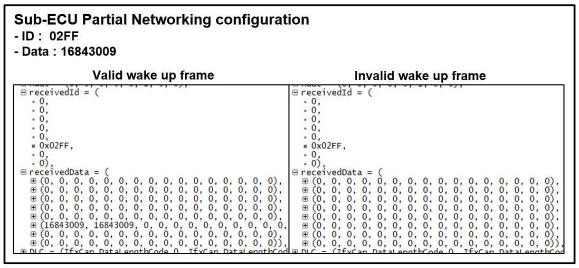

Fig. 18. CAN internal Loop-Back Mode Monitor log

Comparing the three methods mentioned, there is a difference in the minimum number of CAN frames required for initial wake-up process. General CAN communication method requires the 57 frames and other two methods are 2 frames. Current is consumed when each frame is transmitted which is shown in Figure 19. It is usually 18mA when CAN message is not transmitted, but goes up to max 58mA when CAN message is transmitted. In case of General CAN communication method, since 28 times more frames than other methods are transmitted, the current is also consumed by about 28 times.

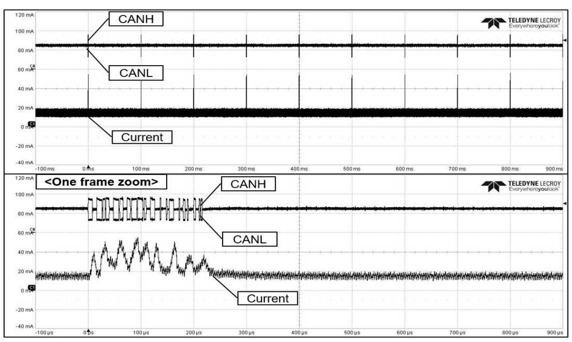

Fig. 19. Current consumption for the CAN frame

A comparision of the three initial wake-up methods is shown in Table 1.

TABLE I. COMPARISION TABLE

	ISO standard process	**DAR function**	**CAN internal Loop-Back**
Minimum number of frames needed to wake up	57	2	2
Advantage	No additional implementation required	Can monitor the error type Can mangage the error counter	No ACK error Can monitor the WUF
Disadvantage	Current consumption is higher than other methods due to min. frames Higher possibility of error warning and bus off	Software has to be implemented	A sufficient number of CAN nodes are required

VI. CONCLUSION

This paper discussed initial wake-up partial networking sequence. As shown in the test results, two methods can be implemented to Partial networking applications. The proposed methods can more efficiently manage the error and reduce current consumption for initial wake-up process.

DEFINITIONS/ABBREVIATIONS

MCU	Microcontroller Unit
WUF	Wake Up Frame
CAN	Controller Area Network
DAR	Disabled Automatic Retransmission
PN	Partial Networking
ECU	Electronic Control Unit
TEC	Transmit Error Counter
REC	Recessive Error Counter
ACK	Acknowledge
ISO	International Organization for Standardization

REFERENCES

[1] Hannah Ritchie “Cars, planes, trains:where do CO2 emissions from transport come from?” in Our Wold in Data

[2] ISO 11898-2:2016 – Road vehicles – Controller area network (CAN) – Part 2 : High-speed medium access unit

[3] Infineon's TLE9255 reference manual, www.infineon.com

[4] ISO 11898-1:2015 – Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling

[5] Infineon's Microcontroller TC39x reference manual, www.infineon.com

[6] Infineon's TLE9461 reference manual, www.infineon.com