ECU with emulated partial networking functionality

An alternative approach to ISO 11898-6 CAN transceivers

Martin Kresta, Roman Buzas, and Ondrej Kupcik, ON Semiconductor

The paper presents a study of an alternative realization of ECU with selective wakeup functionality inspired by the ISO11898-6 standard. Designed ECU enables so-called partial networking, which is one of the recent methods for improving energy efficiency in automotive electronics. The ECU is based on currently available devices used in automotive ECUs including a 16-bit MCU and a DC/DC-based system basis chip. These devices do not provide hardware support for partial networking according to ISO11898-6 and the selective wakeup functionality is then mainly realized by the MCU software. The performed experiments evaluate the power saving potential of such an ECU as well as timing aspects of remote wakeup. The overall ECU consumption of 235uA during bus idle and 3.4mA during ongoing bus activity was measured in selective sleep mode. Furthermore, the consumption during bus activity can be further reduced by optimizing the design of the CAN transceiver IP. Experiments also show that after bus idle, worst case the fourth CAN frame is detected correctly. With respect to those parameters, the proposed solution can be an interesting alternative to the dedicated PN CAN transceivers, especially when cost and EMC performance is considered. The practical limitations of such an approach and their proposed solution are also discussed.

Partial networking introduction

Car manufacturers are looking for ways how to improve fuel efficiency and decrease CO_2 emissions of vehicles. In-vehicle electronic systems are one of the possible areas for an improvement. Especially in modern cars, the electronic systems significantly increase the overall fuel consumption. A substantial amount of energy is wasted by the ECUs which consume power, even if their functionality is not currently needed. A temporary deactivation of such ECUs connected to the in-vehicle network (usually CAN) is called partial networking. Practical applications and expected benefits of this technology were presented for example by Burkhardt [1] or Huber [2].
Partial networking assumes the usage of special CAN transceivers with so-called selective wakeup functionality. In order to define these hardware chips, a draft of standard ISO 11898-6 was created by SWITCH group [2]. The ECUs equipped with ISO 11898-6 transceiver can be put in the so-called selective sleep mode, and remotely woken up by a dedicated CAN message called the Remote Wake-Up Frame (RWUF). This functionality is thoroughly described in [4] and [5]. In selective sleep mode, the MCU is unpowered and CAN messages are processed by the transceiver. Since the PN CAN transceiver has to decode those messages and recognize a valid RWUF, it has to contain a CAN receiver, a precise internal oscillator and comparison logic in addition to the standard CAN transceiver. Samples of these special CAN transceivers are already available.

Alternative solution utilizing a standard CAN transceiver

The fact, that processing of CAN messages in selective sleep mode is done completely by a special transceiver device, brings the benefit of very low power consumption of the ECU. However, the CAN controller receiver IP and precise oscillator are in fact duplicated in the ECU. The CAN controller receiver redundantly embedded in the CAN transceiver chip is used only in selective sleep mode, while in normal mode the CAN controller integrated

in the MCU is used. Furthermore, the realization of a low-power precise internal oscillator with a low cost represents a considerable challenge for chipmakers. With respect to the available information, there is still no device achieving consumption under 500uA during ongoing bus communication, which is the limit required by OEMs [5].
The aim of the following experiment is to emulate partial networking functionality using only a standard CAN transceiver and an MCU, which is the typical case of current automotive ECUs. The goal is to implement a functionality that is as compliant to ISO 11898-6 as possible, while keeping the power consumption reasonably low. This alternative approach assumes the usage of an MCU and an SBC suitable for automotive, low-power applications.

ECU hardware

In order to show and measure the real achieved consumptions, simple automotive ECU was designed and produced. The block diagram of the designed hardware is in figure 1. The core of the ECU is made of an MCU and a system basis chip. Both of these devices provide various low-power operation modes, which make them a suitable choice for low-power applications. The DC/DC voltage regulator of an SBC device allows effective utilization of energy drawn from the battery. Further, the ECU is equipped with a current sensing circuit, which allows observation of a time-variant current, drawn from the VBAT supply, in different operating conditions. As can be seen in figure 1, the evaluation ECU does not contain any specific hardware peripherals. In the real application, those peripherals can stay unpowered in the ECU low-power modes and thus do not influence the overall ECU consumption.

ECU software architecture

As the designed ECU utilizes a standard CAN transceiver, the selective wakeup functionality is implemented by MCU firmware. The complete software stack is inspired by AUTOSAR layered architecture [6], depicted in figure 2.

The functionality of an ISO 11898-6 transceiver is implemented by the Selective Wakeup Manager module. This software module provides APIs for configuration of all necessary parameters in a similar way as the PN transceivers are configured via SPI. The other modules directly involved in partial networking functionality are a CAN driver, a CAN Transceiver driver, an MCU driver and an ECU State manager. Other software modules, especially those in the microcontroller abstraction layer, are based on the AUTOSAR standard in terms of functionality and provided APIs.

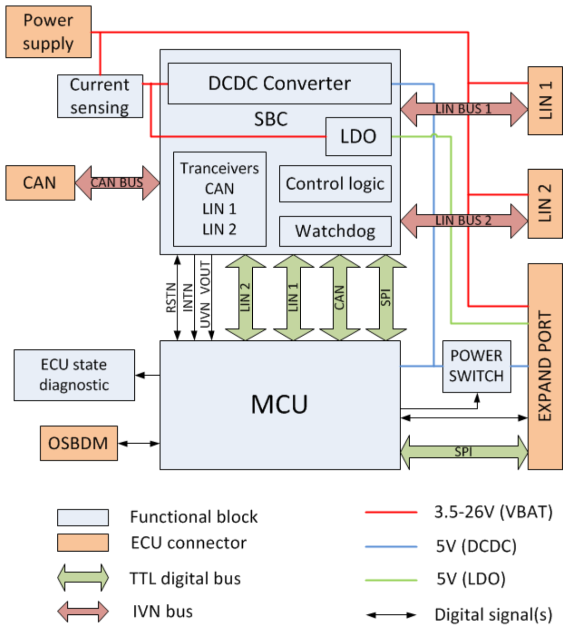

Figure 1: Block diagram of ECU hardware

ECU operating modes

The evaluation ECU supports four different operating modes, which are summarized in table 1 together with their respective current consumptions. The first low-power mode is Standard sleep mode, which supports wakeup sources common for automotive ECUs without PN functionality. The other two low-power modes are added to support partial networking and, in-fact, substitute the selective sleep mode of the ISO 11898-6 transceiver.
Fast-startup sleep mode is similar to the Standard sleep mode. The difference is that a detected CAN wakeup pattern (RWUP) will not cause wakeup to normal mode, but switching to RWUF detection mode.

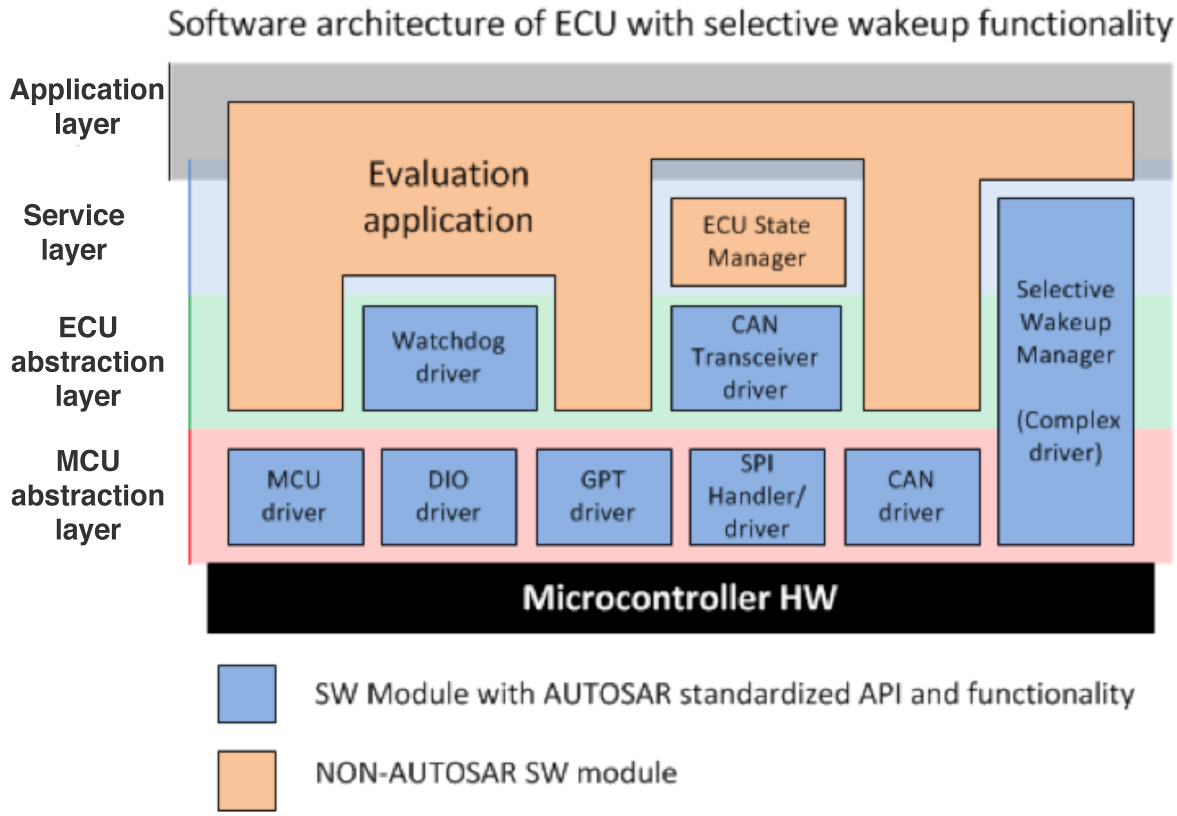

Figure 2: ECU software architecture

In order to achieve short startup time, the MCU is configured to pseudo stop mode, where the MCU oscillator is running, but the clock signal is not distributed neither to ALU nor to MCU peripherals [7].

In the RWUF detection mode, the MCU is configured in wait mode with CAN controller in receive-only mode. If received message passes through ID acceptance filters, the MCU is woken up to run mode in order to check message payload and decide whether the message is valid RWUF. The MCU's system clock SYSCLK is derived directly from an oscillator, and PLL circuit is disabled in this mode. Only the MCU core and CAN controller are enabled, all other MCU peripherals are disabled (disconnected from the system clock) to keep the consumption as low as possible. If the bus becomes idle for a certain time period (denoted as tTOCAN [5]), the ECU automatically switches back to Fast-startup sleep mode. For defining the delay tTOCAN, a low power external timer integrated in the SBC device is used.

Finally, the Normal mode provides full functionality of the ECU. This mode is entered after detection of valid RWUF or other wakeup source (LWU, LIN, etc.).

Evaluation tests

In order to measure time-varying current consumption, various tests were performed during transitions between different operating modes.

The first test simulates the fastest possible selective wakeup scenario. The test also proves that the ECU detects already fourth RWUF correctly (ISO 11898-6 specification requires that maximally the first five CAN frames, following RWUP, can be ignored [4]). The result of this test is shown in figure 3. Initially, the tested ECU is in Fast-startup sleep mode. At time t = 0, another ECU (test master) starts transmission of four identical RWUF frames, with minimal inter-frame space. Those RWUFs use the standard 11bit identifier and empty data field (DLC = 0), which represent the shortest valid CAN frame. At time-point A (see figure 3), the CAN Transceiver (part of SBC) detects wake-up pattern (RWUP) and generates INTN interrupt for MCU. The MCU is woken up by this interrupt and immediately configures the CAN Transceiver and the CAN controller to receive only mode. This configuration is completed at time-point B,

Table 1: ECU operational modes with corresponding static current consumptions

ECU mode Designed ECU power mode	**SBC** Configuration of SBC	**MCU** Configuration of MCU	**Required consumption** by OEM's [5]	**Measured consumption** from VBAT
Standard sleep mode PN disabled	Sleep mode CAN–wakeup	Unpowered	-	60µA (T_A = 25°C)
Fast-startup sleep mode PN enabled	Standby mode CAN–wakeup	Pseudo stop mode Oscillator 4MHz Startup time 3.5µs	30µA 1)	235µA SBC ≈ 100µA MCU ≈ 135µA
RWUF detection mode PN enabled	Standby mode CAN-receiveonly	Wait (Run) mode SYSCLK 4MHz from oscillator	500µA 1)	3.4mA (4.2mA) 2) SBC ≈ 2mA MCU ≈ 1.4mA
Normal mode Full functionality of ECU	Normal mode CAN–normal VOUT–on	Run mode SYSCLK 20MHz from PLL	-	12mA 3) + consumption of functional hardware

Note 1): At this moment no CAN transceiver is known which meets this requirement (5/2013)
Note 2): MCU in Run mode (4.2mA) only when RWUF with matching ID is detected. After decoding of RWUF data, MCU goes back to Wait mode (3.4mA)
Note 3): Measured with CAN bus idle

when the CAN controller starts synchronizing to the CAN bit stream. At time-point C, the CAN controller is synchronized and ready to accept CAN messages. Thus the fourth frame is received and detected as valid RWUF.
The MCU sets the ECU into Normal mode configuration, which is completed at the time-point D. The peak of current consumption (Ic), rising at time of 1ms, is a result of the rapid change of the power needs caused by switching MCU clocks to PLL (4MHz →20MHz) and by setting the SBC into Normal mode.
The next evaluation test captures the current consumption of the ECU in RWUF detection mode during ongoing bus communication. The aim of this test is to show power saving efficiency in configurations when some nodes are in low-power selective sleep mode, while other nodes are using the CAN bus for normal communication. Also, the functionality of tTOCAN delay is demonstrated by this test. The captured data are plotted in figure 4. As well as in the previous test, the ECU is initially in Fast-startup sleep mode.
The test master starts transmission of the CAN messages, simulating 100% bus load. The first transmitted message is treated as a wakeup pattern (RWUP) by sleeping ECU. As a consequence, this ECU enters RWUF Detection mode (time-point A) and its consumption increases to 4.2mA. Since the stream of CAN messages consists of general CAN frames with the ID not matching to configured RWUF, the MCU immediately goes to wait mode, reducing current consumption to 3.4mA. The wait mode is then preserved during the pending communication, because CAN messages do not match with the ID acceptance filter of CAN controller module. At time-point B, the bus communication ends and the bus become idle. The ECU stays in RWUF Detection mode, ready to detect potential RWUF. The external timer generates interrupts every 256ms (time-points C and D). If the bus is idle during the whole period between two consecutive interrupts, the ECU goes back to Fast-startup sleep mode (time-point D). The effective tTOCAN delay is in this example 505ms (measured from time-point B to D).

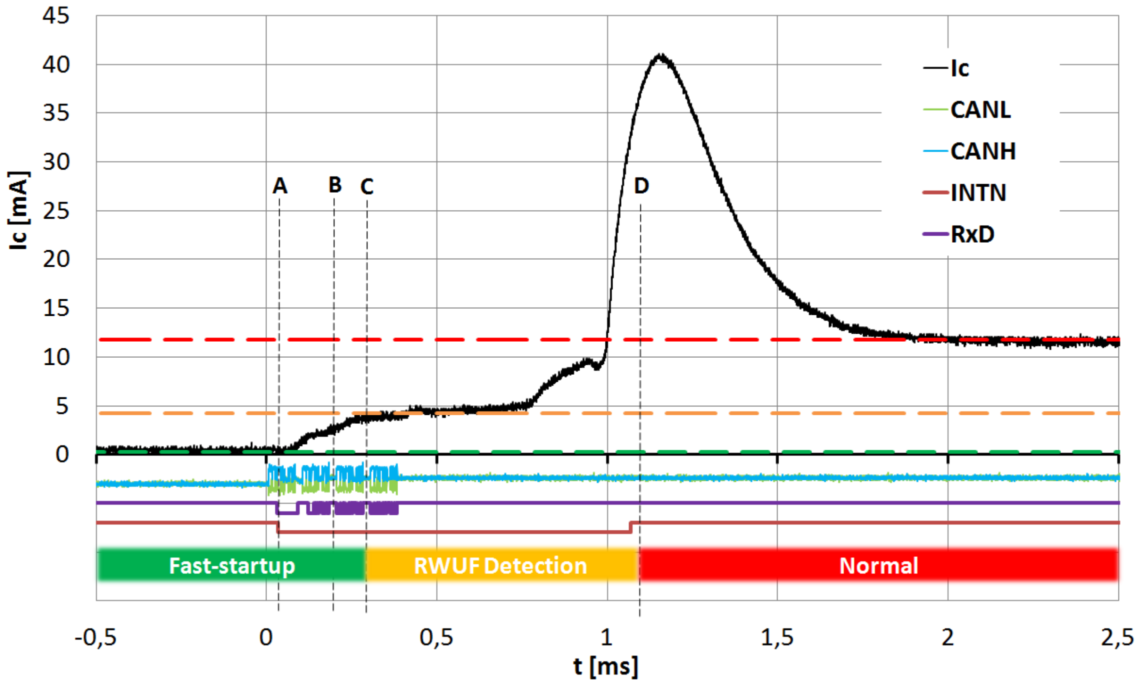

Figure 3: The fastest possible selective wakeup scenario

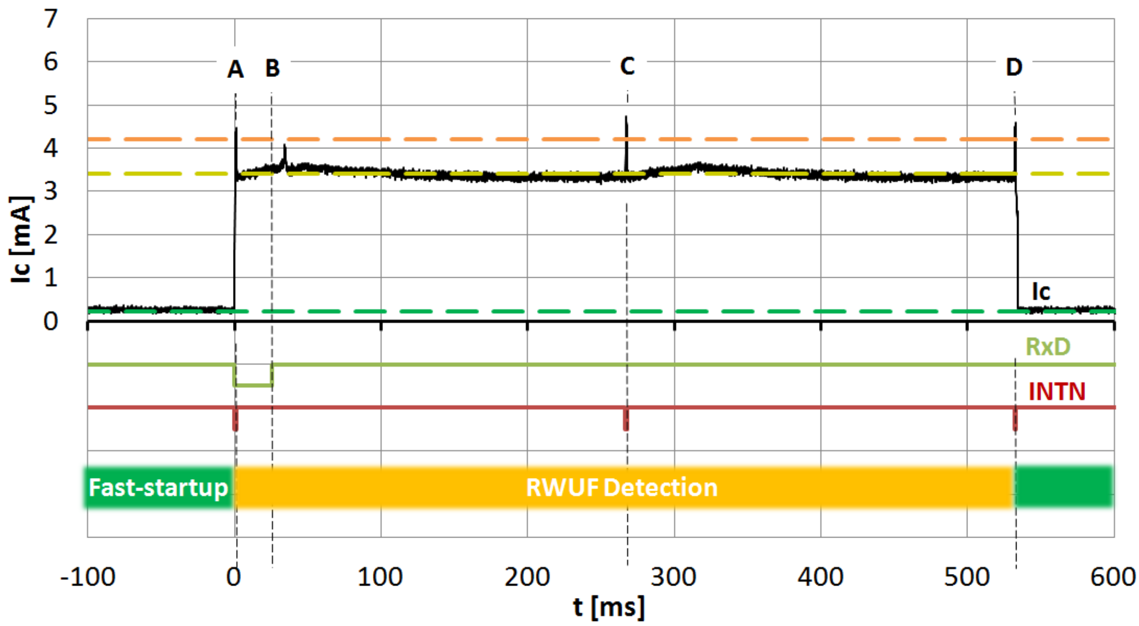

Figure 4: The consumption in RWUF detection mode with 100% bus load

Deviations from ISO 11898-6 standard

The experiment shows, that with used hardware it is not possible to meet all specific requirements of ISO 11898-6 standard.

The first deviation concerns the processing of CAN messages in RWUF Detection mode. The CAN controller module integrated in the MCU accepts only fully valid CAN frames and all frames containing an error are ignored [7]. However, a dedicated PN CAN transceiver should ignore acknowledgement (ACK) and end-of-frame (EOF) errors, when deciding about frame validity. Effectively, this deviation may cause problems when there is only one active node in the CAN network which transmits RWUF frames to wake up other nodes. Since this active node will not receive ACK bit in dedicated ACK slot, its transceiver automatically transmits EOF error, which makes the frame invalid. This corrupted RWUF frame than cannot cause the remote wakeup of other nodes.

Another discrepancy with ISO 11898-6 standard is missing frame error counter mechanism [4]. This mechanism automatically wakes the ECU up when certain number of erroneous CAN frames was detected. Since the CAN controller module does not report erroneous (invalid) CAN frames to the MCU, these events cannot be handled by the software and thus the frame error counter cannot be implemented as required by the standard.

Experiment results summary

The ECU with software realization of selective wakeup functionality was designed as an alternative solution to dedicated PN CAN transceivers. As expected, the current consumption in low-power modes is not as low as with those dedicated devices, however the power saving potential is still substantial, especially thanks to DC/DC converter of used SBC. Furthermore, the effective current consumption of the CAN transceiver in receive-only mode is currently about 2mA (current drawn from VBAT). Newer available CAN transceiver IP is reaching 1mA or even lower current consumption, which can reduce the overall consumption in RWUF detection mode.

As reported in previous chapter, there are a few issues affecting the robustness of selective wakeup functionality preventing usage in real automotive applications. Nevertheless, it was shown that is not necessary to build-in the complete RWUF detection IP into a special transceiver chip in order to achieve reasonably low current consumptions. Instead of that, a CAN controller module integrated in the MCU can be adapted to allow detect RWUFs as required by ISO 11898-6. Also the enhancement of the SBC control logic could significantly reduce the software complexity and allow robust realization of partial networking.

The proposed realization brings some advantages, when compared with dedicated ISO 11898-6 compatible transceivers. Firstly, it uses a precise crystal oscillator and standard CAN transceiver IP with time-proven EMC performance for detection of RWUF. The CAN receiver IP and oscillator are not duplicated in the ECU, which should reduce its overall cost. Secondly, the overall ECU wakeup time from selective sleep mode is faster than with a dedicated ISO 11898-6 transceiver, because the MCU is powered and partially initialized already during the RWUF detection process.

Conclusion

In this paper, an alternative realization of partial networking has been presented. In order to allow real measurements of current consumption, a simple ECU was designed and produced. For the construction of this evaluation ECU, commonly available automotive components were used. The MCU has been equipped with software implementing selective wakeup functionality as described by ISO standard 11898-6. Further, the level of compliancy with this standard has been thoroughly examined. The performed experiments have shown significant power saving potential of created ECU. Also it has been shown that the wakeup time is sufficiently short to meet ISO standard requirements.

Some issues presented in this paper can prevent successful practical usage of this approach. However, minor adaptations of SBC chip and CAN controller IP could address those issues. Specific definition of those necessary adaptations can be considered in future work.

Acknowledgment

This work has been done in the frame of Master thesis at Brno University of Technology. Author therefore thanks to ON Semiconductor in Brno, Czech Republic for support and guidance of this project.

Martin Kresta
ON Semiconductor
Videnska 125
CZ-619 00 Brno
Tel.: +420 547 125 453
martin.kresta@onsemi.com
www.onsemi.com

Roman Buzas
ON Semiconductor
Videnska 125
CZ-619 00 Brno
Tel.: +420 547 125 415
roman.buzas@onsemi.com
www.onsemi.com

Ondrej Kupcik
ON Semiconductor
Videnska 125
CZ-619 00 Brno
Tel.: +420 547 125 468
ondrej.kupcik@onsemi.com
www.onsemi.com

References

[1] Burkhardt, F.: Partial deactivation of CAN nodes. CAN Newsletter.1/2012. CAN in Automation, March 2013. pg. 70-73. B25361

[2] Huber, H.: Partial Networking: In-Vehicle Networks Can Reduce Costs and CO2 Emissions [online]. 2010-3-4 [quote: 2012-25-3]. Available at: http://www.ecnmag.com/articles/2010/03/partial-networking-vehicle-networks-can-reduce-costs-and-co2-emissions

[3] [Butzkamm, M., Bollati, D.: Partial Networking for CAN bus systems: Any saved grams CO2/km is essential to meet stricter EU regulations.[online]. iCC 2012, CAN in Automation.

[4] ISO 11898-6. Road vehicles - Controller area network (CAN) - Part 6: High-speed medium access unit with selective wake-up functionality.

[5] Geneva: International Organization for Standardization, 2012-20-11.

[6] German OEMs: Hardware Require-ments for Partial Networking.

[7] Autosar: Layered Software Archi-tecture [online]. 2010-24-11.Version: 3.1.0. Available at: http://autosar.org/download/R4.0/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[8] Freescale: MC9S12XS256 Reference Manual [online]. 2011-7 Rev. 1.12. Available at: http://cache.freescale.com/files/microcontrollers/doc/ref_manual/MC9S12XS256RMV1.pdf

[9] ON Semiconductor: NCV7471, System Basis Chip with a High-Speed CAN, Two LINs and a Boost-Buck DC/DC Converter. 2012-1-10. Rev. 0. Avail-able at: www.onsemi.com/pub_link/Collateral/NCV7471-D.PDF