Power saving in CAN applications

Magnus-Maria Hell, Infineon Technologies
Ursula Kelling, Infineon Technologies

During recent years, the discussion about power saving had and has different aspects. One of them has been to save power in CAN applications. The mechanisms, which can be used either on physical layer or on the microcontroller, will be discussed within the article.

In a normal CAN network, all CAN nodes are permanently active when the CAN communication is running, independently if the ECU is used or not. However, many applications are not used all the time, and these ECUs can be switched off to reduce the power consumption. There is also a critical situation at every power up. All ECUs ramps up at the same time and this end up in a very high current peak. This can also be reduced, if the necessary ECUs ramps up at the beginning and the other modules will be added if required. A solution for such a realization must fulfill the following criteria's

- No negative impact on the physical bus (no disturbance)
- Can be awoken with a dedicated CAN Frame
- Low current consumption

In addition, three different situations must be covered from this solution. In normal, can communication, in start up phase of the car and in parking cars.

Normal communication mode

In normal CAN communication, not needed ECUs can be set in a special sleep mode. All other CAN nodes can communicate and will be not disturbed from the deactivated ECUs. With a dedicated wake up frame, one or more ECUs can be awoken with a small time-delay.

Start up phase.

During the start of CAN communication, in a CAN network, all ECUs ramp up and together, they will consume a lot of current. This is not necessary and with the new approach, only the needed ECUs should be ramped up. All other nodes changes from sleep mode in a bus observation mode. After the successful ramp up, the other nodes can be added one by one into the communication, if necessary.

Parking cars or standby applications

If a car is parked or a device is in standby mode, a very low current consumption is required, to protect the battery of being unloaded. However, if for example the radio is on, all CAN ECUs located on this CAN bus will stay active and consume a lot of current. With this new solution, only the necessary ECUs are active (for example wheel to control the radio and the radio itself) and all other nodes a sleeping or shut off. This reduces the current consumption dramatically.

Two different solutions cover the power saving criteria.
One, called partial networking is based on a modification of the transceiver and the other one is called pretended networking, which is realized on the microcontroller.

Partial Networking

In the partial networking approach, the wake up frame detection unit is implemented in the high-speed CAN transceiver. This new unit contains

- A high precision oscillator
- A can message decoding unit
- An error handling management
- A Wake up Frame configuration
- A compare unit

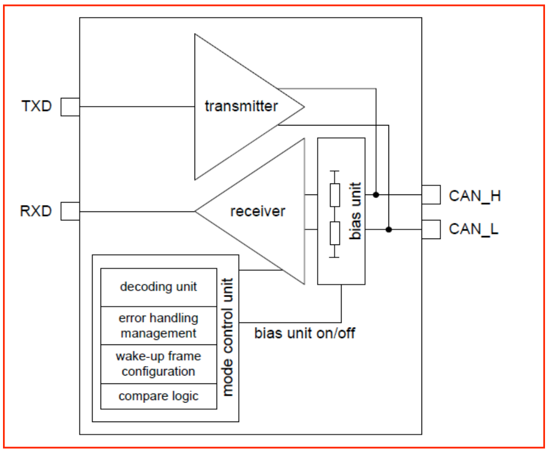

Figure 1: Basic block diagram of a partial networking transceiver

During selective wake mode, this unit is active and monitors the CAN communication like a guard dog. If a dedicated CAN frame is observed, the transceiver wakes up the ECU. These kinds of transceivers have now two modes in the so-called low power mode:

- Sleep Mode and
- Selective wake mode

In sleep mode, the current consumption is reduced to a minimum; all functions in the ECU are disabled. Every message on the bus wakes up the transceiver and the ECU's

In selective wake mode, the current consumption is also low, but the wake up frame detection unit is active, and monitors the bus. All other functions in the transceiver are disabled. With the dedicated wake up frame, the transceiver and the ECU will be woken up.

The advantage from this approach is a very low current consumption in low power mode. The disadvantage is the fact, that only one dedicated Wake-up frame awakes the ECU. In addition, the long ramp up time is a disadvantage for this system. A first implementation with this solution from Infineon can be found on the TLE9267QX. This system base chip includes high-side and low side switches to drive a relays, voltage regulators a LIN Transceiver and a CAN Transceiver with selective wake function to support the partial networking concept. The Wake up Frame can be set via SPI together with a wide range of other functionalities in this product. The low current consumption high precision RC Oscillator with a long time stability of less than 1% guarantees a reliable receiver performance over lifetime of this product. In this product, the can detection unit is the receiver part of the can communication controller acc. to ISO 11898-1. This allows a very stable receiver performance as known in the established microcontrollers. If the unit cannot detect the running frames for example due to bad signal integrity on the bus, the unit sets a failure flag after 32 not correctly detected frames. To reduce the noise impact, a failure counter is implemented. If a frame could not be not detected the counter increase and a correct detected frames reduce the counter reading. If the transceiver is in selective wake mode and the counter reading is above 32, the transceiver changes into standby mode and wakes up the voltage regulator and the microcontroller. This allows using the can controller in the microcontroller. The precision of the crystal oscillator is higher and can receive the messages with the worse signal integrity.

If the SBC is in selective wake mode, the microcontroller is off, and it could be now the situation, that the bus communication stops. To reduce the current consumption, the SBC changes into the normal sleep mode if no bus communication is observed for more than 1 second. When the bus communication is restarted, the SBC changes into selective wake mode. After 500µs, the oscillator is synchronized again and the SBC selective wake detection unit monitors the bus. The feature allows reducing the peak current after a power up of the car or the system. First, the mandatory units will be woken up and after the end of the ramp up face, additional ECUs can be added. With this feature, the ramp up peak current can be reduced. For this feature, it is required, that the selective wake SBCs is supplied permanently. This feature cannot be used if the power of the SBC was switched off. After every power up the SBC, the device must be re-initialized.

2.5V Biasing

If the bus communication is finished, all transceivers have changed into sleep mode and the bus is terminated with the input resistors by every transceiver to ground.
If the bus communication starts again, the transceiver changes its termination to VCC/s (called Biasing) if the transceiver is set into normal mode. The ramp up time for every ECU is different, so the termination will not be changed at the same time. It will be changed in a random manner. This has an impact on the signal integrity during this ramp up phase, and error frames can occur. To improve this behavior, the biasing handling is changed. After a wake up of a transceiver, the biasing will be switched on directly and the termination will be set to the biasing. This improves the signal integrity after ramp up significantly. Do be robust against disturbance on the bus of pulses on the bus, the handling was describes in the ISO 11898-6. A dominant pulse on the bus longer than 5µs changes the internal mode from Ini(t) (bias off) to one (bias off). If a recessive level on the bus follows for a period longer than 5µs the mode changes to step2 (bias off) after the second dominant pulse on the bus, the biasing switches on, the transceiver wakes up and changes into standby mode. If the transceiver stays in mode step 1 or 2 for longer than t_{wake}, the transceiver falls back to mode step Ini(t) and reduces the current consumption to a minimum.

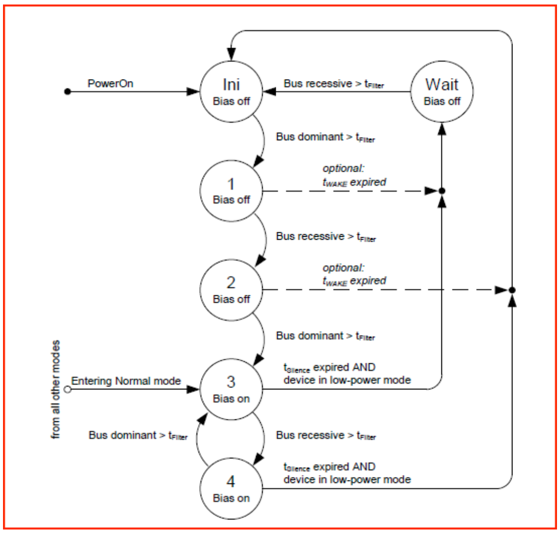

Figure 2: Statediagram for WUF detection

Bit decoding

To allow a wide spread of implementation in inexpensive technologies, no sample point adjustments are specified in the ISO 11898-6. In the ISO constrains for the network are now specified, but these are the minimum requirements and this parameter can be expanded from the different semiconductor manufacturer. Two different scenarios are specified within ISO.
Scenario A: The bit stream consists out of multiple instances of the signal shape A to handle ringing in the network at the end of the dominant level.
Scenario B: The bit stream can be assembled out of multiple instances of the signal shape B1 and one instance of the signal shape B2, to handle the oscillator clock tolerance from the transmitting and receiving unit and handle loss of arbitration. These two types of signal are shown in the next Figure.

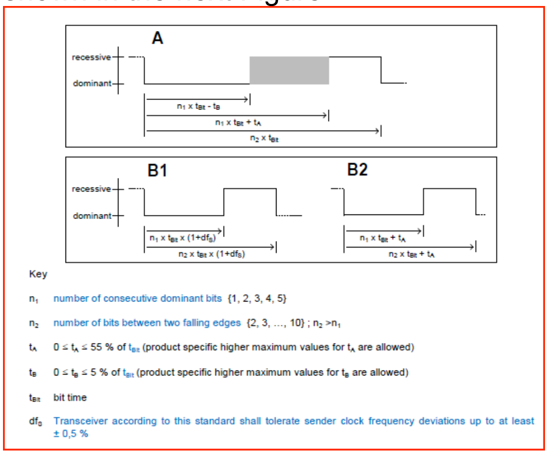

Figure 3: Bit decoding

Scenario A includes the network propagation delay during arbitration and the ringing. The network must fulfill these requirements.
Scenario B1 and B2 has an impact on the transmitting oscillator tolerance. Less than 0.5% is recommended to guarantee a reliable communication.

Partial networking gives the possibility to save the maximum of power, but some disadvantages exist. First, the microcontroller is shut off; therefore, the microcontroller has to run through the complete reset sequence. Therefore, it takes up to 100ms for a non-safety controller and up to 200ms for a safety controller to be active on the bus again. Second, the wakeup message will be lost. From a system point of view, the partial networking concept needs a so-called power master; the ECU is responsible for the wakeup procedure. Systems which are not allowed to have one of the three items, can still save power on a CAN network.

Pretended Networking

Pretended networking describes an approach developed in AutoSAR. Pretended networking in combination with ECU degradation allows to safe power on microcontroller basis. What do these concepts mean? First, they integrate well-known power saving approaches back into AutoSAR, possibly still used by the industrial world.

ECU degradation allows using the HALT mode, also known as IDLE for CPUs. In case of no task is running, the CPU is no longer doing NOPs, but will go to IDLE and therefore the CPU is no longer clocked. As soon as the first interrupt is executed, the CPU is active again. This is already the first power saving step. The next power saving step, is that all modules not used in a low power operation mode, will be shut off, therefore again, power can be saved. By switching off peripherals of the microcontroller, also external devices might be powered off. Now going for pretended networking, the communication modules will run with a reduced message catalogue and reduced amount of interrupt sources. Therefore, less interrupts occur and therefore longer sleeping times of the CPU are possible. These well-known power saving measures, where initially used only during the park situation of a car. It is new that they will now also be used during driving. Dependent on the existing software stacks, these measures will be well known, or can be integrated as long as the software is able to cope with shut off modules or CPU in idle. These mechanisms are sophisticated, as for example they are only existing on devices as the C167 and have been part of newer devices ever since. An additional measure, which needs more software influence is the changing of the clock for a device. In case the device is not prepared for such a measure, all communications have to be stopped, the baudrates have to be adjusted, and then all communications can be restarted. If microcontrollers are prepared, than a central clock switch will exist, not touching any communication. If all measures are combined, saving of 50-60% of the ECUs power consumption can be achieved. Even though this might sound neglectable in times, where every mA counts, these measures will help to safe a sufficient amount of power.

For industrial applications where less software standardization took place, it might be even easier to integrate these simple power saving mechanisms into software.

To support pretended networking on hardware level, the recently announced Aurix family got new features within the MultiCAN+ module. MultiCAN+ is the well known MultiCAN module with some additional features.

Within the CAN module, it is possible to set an interrupt source on every single message object and to reconfigure them during runtime. New to the module is that the so called network management (heartbeat of the network) can continue to run, without any software influence. Up to three messages can be triggered automatically in equidistant time slots. Supporting that the ECU continues to look awake from the outside, even though, it is currently in idle mode. Otherwise, it is necessary that in case the network management is stopped, that the ECU realizes that the network management is no longer active. Therefore a receive timeout is necessary. This receive timeout can be connected to the message objects. If a message is received by one of the chosen message objects, the timer is restarted. If no message is received, the timer will underflow and issue a receive timeout interrupt.

Pretended networking provides by these simple mechanisms an approach which enables the application to keep the power in IDLE mode as long as possible. With Infineon microcontrollers, the CPU is out of IDLE as soon as the interrupt is issued. The disadvantage of the boot-up time of a partial networking ECU does not exist. In addition, the wakeup messages are saved within the CAN module, so the wakeup source can be identified. If more wakeup messages were received, also this can be identified. The latest messages received during sleep mode, are still available for the application.

Summary

The transceiver as well as the microcontroller can help to safe power. For all applications, which have wake up times greater than 100ms or 200ms for safety applications, it is possible to shut off the ECU via the transceiver as long as the network is able to handle this. For all others pretended networking in combination with ECU degradation will help to safe as much power as possible. Many of the features already exist in today's microcontrollers, as for example the XC2000/XE166 or Audo MAX family. With MultiCAN+ on the AURIX™ family, additional features have been introduced, helping the ECU to stay longer in a low power mode.

Magnus-Maria Hell
Infineon Technologies AG
Am Campeon 1-12
85579 Neubiberg, Germany
Magnus-Maria.Hell@infineon.com
www.infineon.com

Ursula Kelling
Infineon Technologies AG
Am Campeon 1-12
85579 Neubiberg, Germany
Ursula.Kelling@infineon.com
www.infineon.com