1.0 Overview

There are a few known use cases where it can be required to enter sleep mode as a result of a CAN wakeup. In this application note these use cases are briefly described, then restrictions of the CANbedded architecture are explained and finally a possible solution is outlined.

2.0 Use cases

2.1 Local wakeup via CAN transceiver

Typically a local wakeup event is signaled to the micro controller via an external interrupt input. Some CAN transceivers provide an external wakeup input which can be used if the micro controller has no external interrupt inputs or if they are already in use. In such a hardware setup the CAN driver cannot distinguish between a wakeup event caused by a received CAN frame or a local wakeup. Therefore some special processing is required by the application. In such a configuration it could happen that as a result of a local wakeup no bus communication is required. This means that the CAN controller which has just been woken up has to be set back to sleep mode again.

2.2 Filtering of wakeup interrupts

Some OEMs require to filter out wakeup events which are caused by noise on the bus. In such a scenario the wakeup event has to be delayed until the reception of a CAN frame is detected by the CAN driver. If the wakeup was caused by noise on the bus the CAN controller has to be set to sleep mode again.
3.0 Limitations of CANbedded

The CANbedded architecture does not allow to call the CAN driver API CanSleep() directly in the context of the CAN driver callback ApplCanWakeUp(). Because CanSleep() is also called immediately if the OSEK NM is initialized with parameter NM_CANSLEEP it is also not possible to call NmOsekInit(NM_CANSLEEP) in ApplCanWakeUp().

Note: Please note that NmOsekInit(NM_SLEEPIND) does not cause an immediate call to CanSleep(), therefore this commonly used sequence is not affected by the limitation.

In the second scenario (filtering of wakeup events) there is a time window between the occurrence of the wakeup and the call of CanSleep() / NmOsekInit(NM_CANSLEEP), therefore this scenario is not affected by the restriction and will not be described further.

4.0 Possible solutions

4.1 Local wakeup via CAN transceiver

4.1.1 With CCL

If the CCL layer is used the wakeup handling is done automatically within CCL and not by the application. The CCL layer does currently not support such a wakeup concept. If you use CCL and plan to use such a wakeup concept please contact Vector.

4.1.2 Without CCL

If the wakeup event is handled by the application and if it is determined in the ApplCanWakeUp() function that it was a local wakeup and bus communication is not required a flag shall be set which is evaluated later on task level.

The following pseudo C-code fragment shows an example:

```c
void ApplCanWakeUp(void) /* CAN driver wakeup interrupt */
{
...
  if(TransceiverWakeupReason==local)
  {
    if(BusCommRequired==TRUE)
    {
      /*Init OSEK NM*/
      NmOsekInit( NM_SLEEPIND );
    }
    else
    {
      /* set flag that sleep mode shall be entered again */
      SleepFlag = TRUE;
    }
  }
  else
  {
    /*Init OSEK NM*/
    NmOsekInit( NM_SLEEPIND );
  }
  ...
}

void ApplTask_X(void) /* handling of sleep mode after local wakeup on task level */
```
The pseudo variables TransceiverWakeupReason, BusCommRequired and SleepFlag are provided and handled by the application. If the SleepFlag is part of a bit field the application has to make sure that flag accesses are performed atomically. In case of a system with multiple channels the described procedure is required for each channel with sleep support where a local wakeup via CAN transceiver is possible.

When the procedure is applied, it could happen that between the local wakeup event and the processing of the flag on task level another ECU in the network wakes up and sends its NM message. In such a case three different ways to proceed are possible:

If a small delay of the ECU wakeup is acceptable no special processing is necessary. The message received between local wakeup and flag processing on task level will be lost.

If a small delay is not acceptable and NM messages are always the first messages on the bus the following code in the NM callback ApplNmCanNormal is required:

```c
void ApplNmCanNormal(void)
{
    if (SleepFlag == TRUE) {
        SleepFlag = FALSE;
    }
}
```

If a small delay is not acceptable and application messages can appear before NM messages on the bus the following code in the CAN driver callback ApplCanMsgReceived is required:

```c
vuint8 ApplCanMsgReceived(CanRxInfoStructPtr rxStruct)
{
    if (SleepFlag == TRUE) {
        SleepFlag = FALSE;
        NmOsekInit(NM_SLEEPIND);
    }
    return kCanCopyData;
}
```

The SleepFlag has to be initialized with FALSE by the application when the CANbedded stack is initialized e.g. after power-on reset.

5.0 Additional Resources

6.0 Contacts

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
<th>Fax Number</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector Informatik GmbH</td>
<td>Ingersheimer Straße 24, 70499 Stuttgart, Germany&lt;br&gt;Tel.: +49 711-80670-0&lt;br&gt;Fax: +49 711-80670-111&lt;br&gt;Email: <a href="mailto:info@vector-informatik.de">info@vector-informatik.de</a></td>
<td>Tel.: +49 711-80670-0&lt;br&gt;Fax: +49 711-80670-111&lt;br&gt;Email: <a href="mailto:info@vector-informatik.de">info@vector-informatik.de</a></td>
<td>Tel.: +49 711-80670-0&lt;br&gt;Fax: +49 711-80670-111&lt;br&gt;Email: <a href="mailto:info@vector-informatik.de">info@vector-informatik.de</a></td>
<td>Tel.: +49 711-80670-0&lt;br&gt;Fax: +49 711-80670-111&lt;br&gt;Email: <a href="mailto:info@vector-informatik.de">info@vector-informatik.de</a></td>
</tr>
<tr>
<td>Vector CANtech, Inc.</td>
<td>39500 Orchard Hill Pl., Ste 550, Novi, MI 48375&lt;br&gt;Tel.: +1-248-449-9290&lt;br&gt;Fax: +1-248-449-9704&lt;br&gt;Email: <a href="mailto:info@vector-cantech.com">info@vector-cantech.com</a></td>
<td>Tel.: +1-248-449-9290&lt;br&gt;Fax: +1-248-449-9704&lt;br&gt;Email: <a href="mailto:info@vector-cantech.com">info@vector-cantech.com</a></td>
<td>Tel.: +1-248-449-9290&lt;br&gt;Fax: +1-248-449-9704&lt;br&gt;Email: <a href="mailto:info@vector-cantech.com">info@vector-cantech.com</a></td>
<td>Tel.: +1-248-449-9290&lt;br&gt;Fax: +1-248-449-9704&lt;br&gt;Email: <a href="mailto:info@vector-cantech.com">info@vector-cantech.com</a></td>
</tr>
<tr>
<td>VecScan AB</td>
<td>Lindholmspiren 5, 402 78 Göteborg, Sweden&lt;br&gt;Tel.: +46 (0)31 764 76 00&lt;br&gt;Fax: +46 (0)31 764 76 19&lt;br&gt;Email: <a href="mailto:info@vecscan.com">info@vecscan.com</a></td>
<td>Tel.: +46 (0)31 764 76 00&lt;br&gt;Fax: +46 (0)31 764 76 19&lt;br&gt;Email: <a href="mailto:info@vecscan.com">info@vecscan.com</a></td>
<td>Tel.: +46 (0)31 764 76 00&lt;br&gt;Fax: +46 (0)31 764 76 19&lt;br&gt;Email: <a href="mailto:info@vecscan.com">info@vecscan.com</a></td>
<td>Tel.: +46 (0)31 764 76 00&lt;br&gt;Fax: +46 (0)31 764 76 19&lt;br&gt;Email: <a href="mailto:info@vecscan.com">info@vecscan.com</a></td>
</tr>
<tr>
<td>Vector France SAS</td>
<td>168 Boulevard Camélinat, 92240 Malakoff, France&lt;br&gt;Tel.: +33 (0)1 42 31 40 00&lt;br&gt;Fax: +33 (0)1 42 31 40 09&lt;br&gt;Email: <a href="mailto:information@vector-france.fr">information@vector-france.fr</a></td>
<td>Tel.: +33 (0)1 42 31 40 00&lt;br&gt;Fax: +33 (0)1 42 31 40 09&lt;br&gt;Email: <a href="mailto:information@vector-france.fr">information@vector-france.fr</a></td>
<td>Tel.: +33 (0)1 42 31 40 00&lt;br&gt;Fax: +33 (0)1 42 31 40 09&lt;br&gt;Email: <a href="mailto:information@vector-france.fr">information@vector-france.fr</a></td>
<td>Tel.: +33 (0)1 42 31 40 00&lt;br&gt;Fax: +33 (0)1 42 31 40 09&lt;br&gt;Email: <a href="mailto:information@vector-france.fr">information@vector-france.fr</a></td>
</tr>
<tr>
<td>Vector Japan Co. Ltd.</td>
<td>Seafort Square Center Bld. 18F, 2-3-12, Higashi-shinagawa, Shinagawa-ku, J-140-0002 Tokyo&lt;br&gt;Tel.: +81 3 5769 6970&lt;br&gt;Fax: +81 3 5769 6975&lt;br&gt;Email: <a href="mailto:info@vector-japan.co.jp">info@vector-japan.co.jp</a></td>
<td>Tel.: +81 3 5769 6970&lt;br&gt;Fax: +81 3 5769 6975&lt;br&gt;Email: <a href="mailto:info@vector-japan.co.jp">info@vector-japan.co.jp</a></td>
<td>Tel.: +81 3 5769 6970&lt;br&gt;Fax: +81 3 5769 6975&lt;br&gt;Email: <a href="mailto:info@vector-japan.co.jp">info@vector-japan.co.jp</a></td>
<td>Tel.: +81 3 5769 6970&lt;br&gt;Fax: +81 3 5769 6975&lt;br&gt;Email: <a href="mailto:info@vector-japan.co.jp">info@vector-japan.co.jp</a></td>
</tr>
</tbody>
</table>