APPLICATION NOTE

Sleep Mode Handling of the ATA6670 Dual LIN Transceiver

ATA6670

Introduction

The Atmel® ATA6670 is a fully integrated Dual LIN transceiver designed to handle low-speed data communication in vehicles. There are two completely independent and separated LIN transceivers Atmel ATA6663 integrated in one package (only the GND pins GND1 and GND2 are internally connected), each featuring three different operating modes: Normal mode, Sleep mode, and Fail-safe mode.

In this application note it is assumed that the INH pin of transceiver 2 does not switch the power supply of the microcontroller via an external voltage regulator. In LIN master nodes, for which the Atmel ATA6670 is primarily designed, the power supply of the microcontroller is normally not switched off completely if there is no communication on the connected LIN networks.

This document describes how to securely switch each of the two LIN transceivers to the different operating modes (especially Sleep mode), including basic code examples. These code examples assume that the part-specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header file. The code examples are limited to the Atmel AVR® microcontrollers and the IAR™ C and GCC compilers.

The two LIN transceivers are nearly identical; the only difference is an additional wake input and an INH output at transceiver 2. This INH output can be used for measuring the battery voltage.

1. Operating Modes

Figure 1-1 shows the operating modes of each LIN transceiver inside the Atmel® ATA6670 as well as the transitions between them. The LIN transceiver can only be switched to Sleep mode from Normal mode and can then be woken up by various events from Sleep mode:

- LIN bus (remote wake-up)
- EN pin
- WAKE2 pin (local wake-up, only transceiver2)
- Vs undervoltage

Figure 1-1. Atmel ATA6670 - Operating Modes

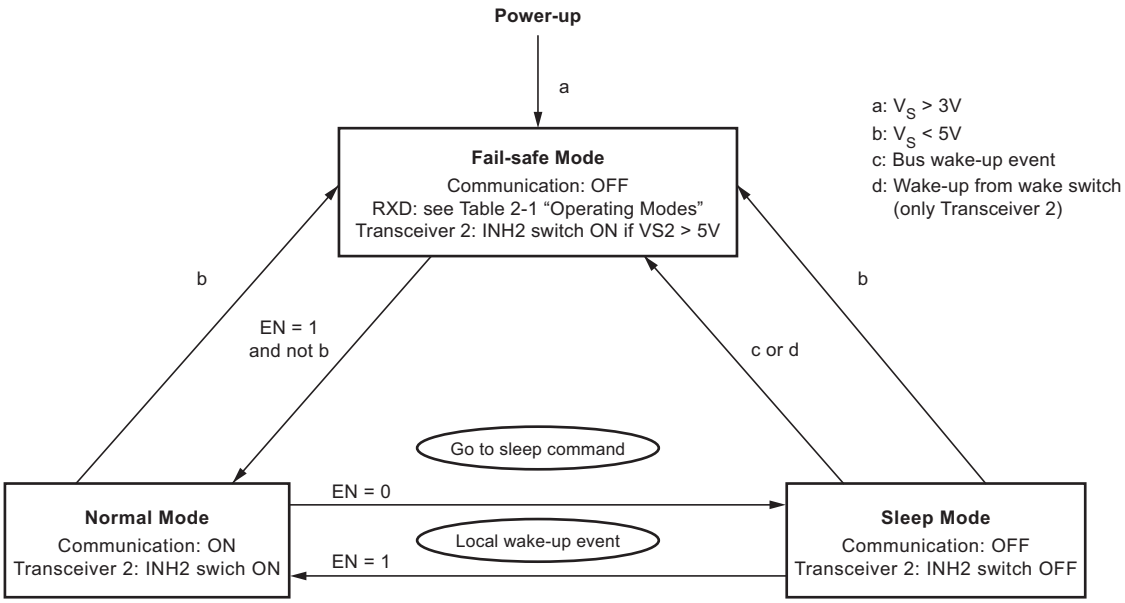


Table 1-1. Operating Modes

Operating Mode	Transceiver 1, 2	INH2	RXD 1, 2	LIN 1, 2
Fail-safe	Off	On, except VS2 < 5V	High, except after wake-up	Recessive
Normal	On	On	LIN-dependent	TXD-dependent
Sleep	Off	Off	High ohmic	Recessive

1.1 Putting the Atmel ATA6670 LIN Transceivers into Sleep Mode

In order to switch the LIN transceivers into Sleep mode the corresponding EN pin has to be pulled to GND and TXD has to remain at high level. After the time T_{sleep} beginning directly at the falling edge of the EN signal has expired, the transceiver enters Sleep mode. The INH2 output is switched off when the LIN transceiver 2 enters Sleep mode.

Figure 1-2. Switching to Sleep Mode

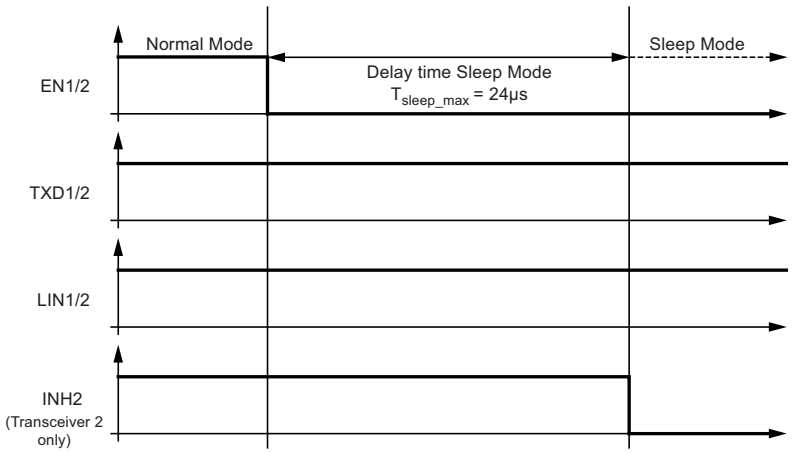
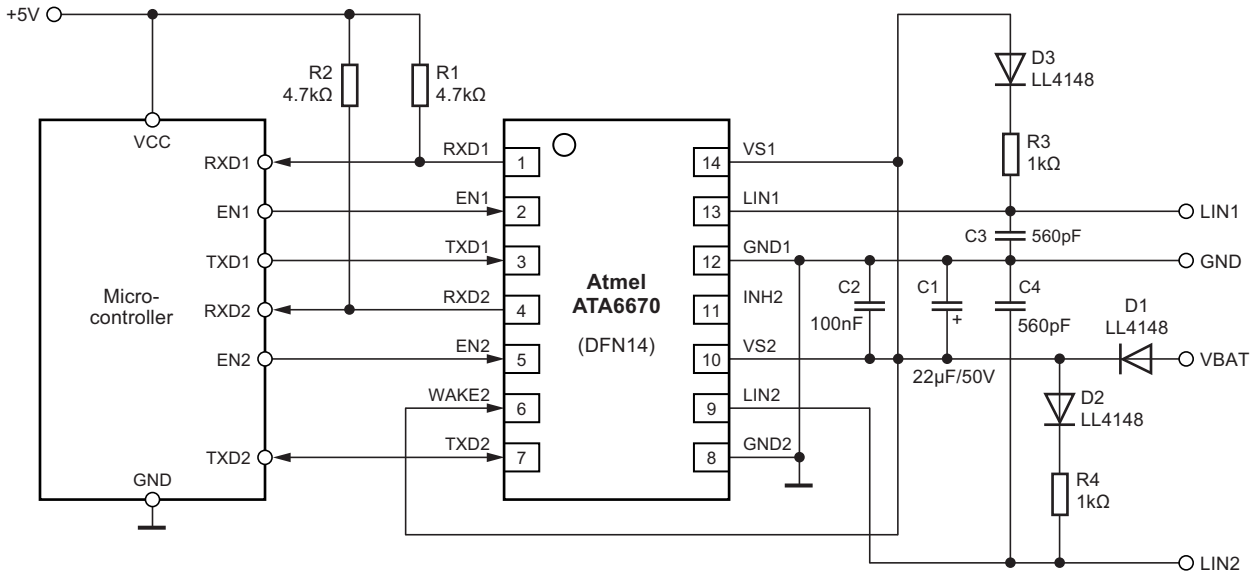


Figure 1-3. Application Schematic



There are two common ways to wake up transceiver 2: a remote wake-up via LIN2 or a local wake-up via the WAKE2 pin. But the transceiver 2 also wakes up in the event of an undervoltage at pin VS2 or at a rising edge at the EN2 pin. The same wake-up possibilities exist for transceiver 1, but without local wake-up because transceiver 1 does not have a WAKE pin. In all cases the LIN transceivers switch from Sleep mode to Fail-safe mode. To avoid deadlock situations the microcontroller must know exactly in which state the transceivers are at all times. Note that the INH2 pin and the WAKE2 pin are not used in the schematic in Figure 1-3.

1.2 Switching to Sleep Mode without any further precaution

1.2.1 Rudimentary C Routine for Switching an Atmel ATA6670 LIN Transceiver into Sleep Mode

Every LIN transceiver inside the Atmel® ATA6670 can be put into Sleep mode using this simple C function:

```
unsigned char Switch_ATA6670_To_SleepMode (void)
{
      // Before setting the EN pin to low it has to be sure, that the TXD line
      // remains high. Therefore the TXD pin has to be set to output high here
      // at the latest. The output high will be overwritten by the UART as long
      // as the UART transmitter is enabled.
      // Consequently, the UART transmitter must be disabled.
      TXDPIN_1_high;            // Set the TXD pin @ the microcontroller to
                                // output high
      TXDPIN_1_output;          // Set the TXD pin to output in the DDRx
                                // register
      UCSRB_1 &= ~(1<<TXEN_1);  // Disable the UART transmitter

      TXDPIN_2_high;            // Set the TXD pin @ the microcontroller to
                                // output high
      TXDPIN_2_output;          // Set the TXD pin to output in the DDRx
                                // register
      UCSRB_2 &= ~(1<<TXEN_2);  // Disable the UART transmitter

      // Now the actual switching of the ATA6670 into Sleep mode takes place
      ENABLEPIN1_low;           // Switch EN1(LIN transceiver 1) pin to low;
                                // assuming that the microcontroller pin is
                                // already set to output
      ENABLEPIN2_low;           // Switch EN2(LIN transceiver 2) pin to low;
      return 1;
}
```

1.3 Possible Issues without Using any Safety Precautions

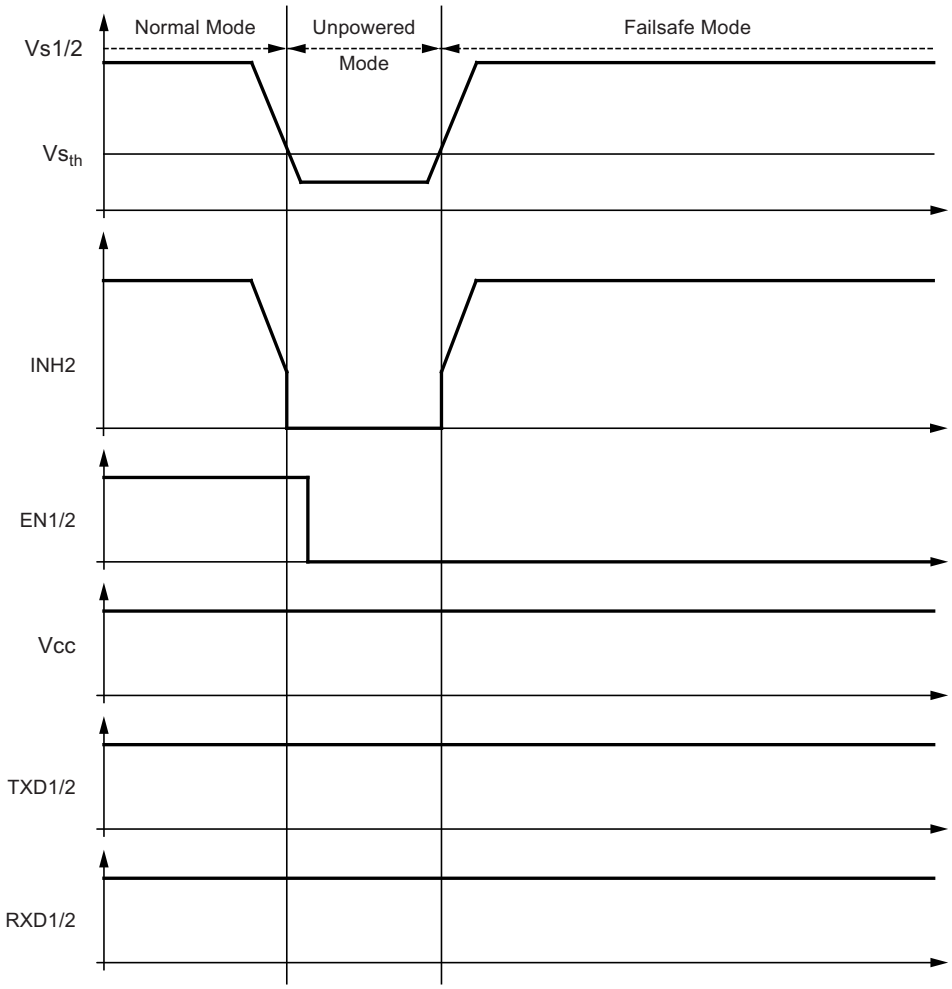
There is one scenario which could lead to a situation where the connected microcontroller tries to switch the Atmel ATA6670 LIN transceivers to Sleep mode; however, the corresponding “Go to sleep” command will not be successful. This is the case if the “Go to sleep” command occurs during a Vs undervoltage condition.

If an undervoltage event occurs while switching into Sleep mode, using the rudimentary C routine from Section 1.2.1 on page 4 may cause a module deadlock. An undervoltage event is triggered when the voltage at the VS pin of the LIN transceiver falls below 5V. In this case the Atmel ATA6670 LIN transceivers automatically switch to Unpowered mode without signaling the mode change to the microcontroller. In general, the Unpowered mode is the same as the Fail-safe mode, but the LIN transceivers do not change their state according to the corresponding EN pin; the LIN transceivers therefore do not go into Sleep mode.

If the microcontroller does not change the state of the EN pin (the EN pin remains high) during this period, then the LIN transceivers will automatically re-enter Normal mode after the battery voltage has recovered. In compliance with the LIN specification, no data transmission via LIN is possible during the undervoltage period.

If the LIN transceivers detect Vs undervoltage and have changed to Unpowered mode, any further mode change via the EN pin will not lead to any action. The INH2 output is switched off during Unpowered mode. As the microcontroller has no information about the LIN transceivers being in the Unpowered mode, it may attempt to switch the LIN transceivers to Sleep mode. This scenario is depicted for transceiver 2 in Figure 1-4 on page 5.

Figure 1-4. Undervoltage When Switching to Sleep Mode



After the undervoltage at VS2 disappears, the LIN transceiver2 switches from Unpowered mode to Fail-safe mode and as a result the INH2 output switches on again. The LIN transceiver2 stays in Fail-safe mode because the EN2 pin is still low. But the microcontroller expects that the LIN transceiver2 has been gone into Sleep mode. As a result, the microcontroller does not even consider putting the LIN transceiver into Normal mode via the EN2 pin. The transceiver does not respond to any wake-up events, because it is not in Sleep mode, but instead in Fail-safe mode. But the microcontroller expects the transceiver to be in Sleep mode and the module remains in a deadlock, isolated from LIN communication.

2. How to Avoid Potential Deadlocks

Because the Atmel® ATA6670 is used primarily in master applications, the microcontroller is permanently supplied independent from the Atmel ATA6670. The microcontroller expects the LIN transceivers to be in Sleep mode and therefore ready for a wake-up, but the wake-up will never work because the transceivers are already in Fail-safe mode and the module in a deadlock.

This scenario can be avoided by either a hardware solution or a software solution.

2.1 Hardware Solution – Monitoring the Supply Voltage Via the INH2 Output

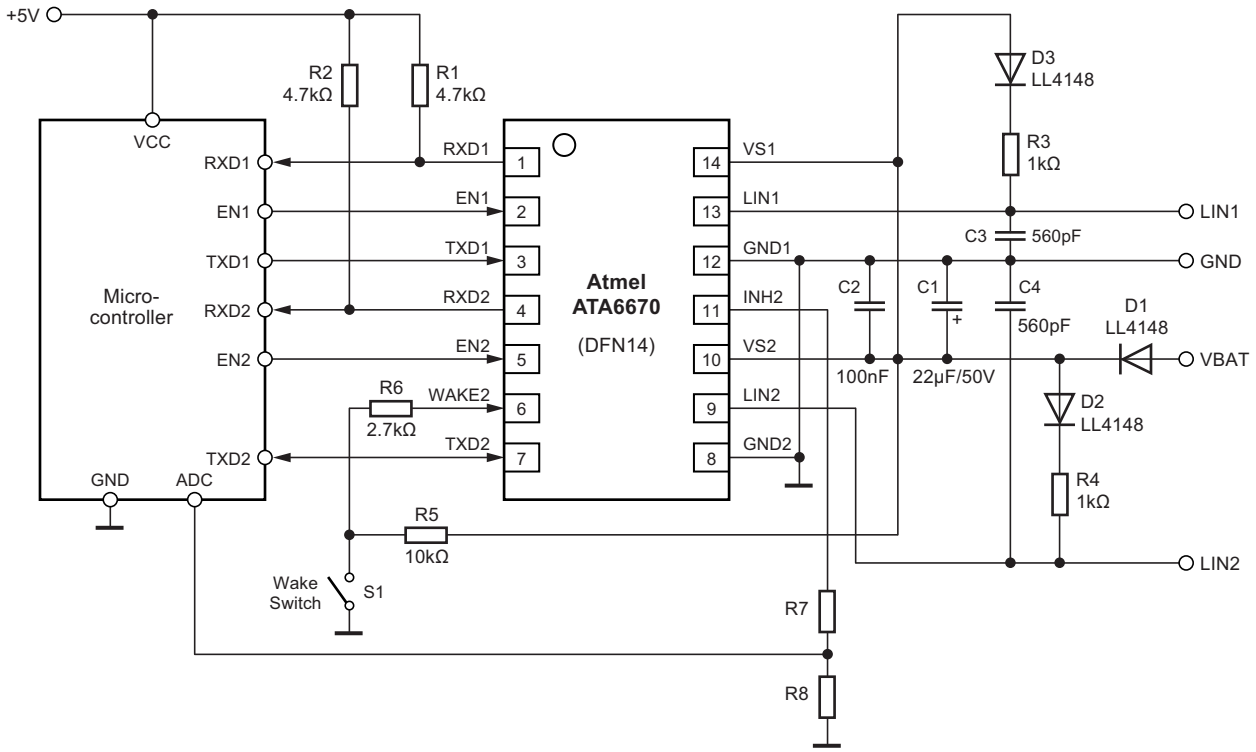
For this solution it is assumed that the supply pins VS1 and VS2 are interconnected (see Figure 2-1 on page 7). It is possible to monitor the VS2 supply voltage with the help of the INH2 output. But because the INH2 pin is a battery-related output, special care must be taken when connecting this pin to the microcontroller. For this reason, an additional voltage divider between the INH2 output pin and ground is necessary. This limits the voltage level at the microcontroller input to noncritical values. This voltage divider can be connected to an A/D converter input of the microcontroller, giving the microcontroller the actual voltage level of the VS2 pin. This is shown in Figure 2-1 on page 7. If the actual voltage level is not important for the microcontroller, it might be better to connect the INH2 pin via a voltage divider to an internal analog comparator input of the microcontroller. It is easy to set the threshold at which the Atmel ATA6670 power supply is sufficient by the dimension of the voltage divider.

In both cases the module's current consumption is not increased in Sleep mode, because the INH2 output is switched off in Sleep mode, keeping the voltage divider inactive in this mode.

Because each of the two independent Atmel ATA6670 LIN transceivers has separate manufacturing tolerances, they will most likely have two slightly different Vs undervoltage thresholds. This means if both LIN transceivers are in Sleep mode and the battery voltage decreases, it is possible that the INH2 output switches on because the undervoltage threshold Vsth of the second LIN transceiver is reached while the LIN transceiver 1 is still in Sleep mode. The Vs undervoltage threshold Vsth can vary between 4.0V and 5.0V. This means that it is only possible to precisely monitor the operating mode of transceiver 2 with the INH2 output, but not monitor transceiver 1. Therefore we recommend defining an undervoltage threshold at the ADC that covers all Vs undervoltage situations for both transceivers, meaning at supply voltages below 5V switching to Sleep mode must be avoided in order to be sure that after the “Go to sleep” command none of the LIN transceivers is in Fail-safe mode instead of Sleep mode.

A second solution is to cyclically switch the transceiver 1 into Normal mode and then back into Sleep mode in order to be sure that the device is in Sleep mode at least most of the time while ensuring that no deadlock situation can occur. This software solution, as described in the following section, is also applicable for transceiver 2 if it is not possible to monitor the battery voltage as described above.

Figure 2-1. Monitoring the Supply Voltage via the INH2 Pin



2.2 Software Solution

If the microcontroller is supplied permanently and independently from the Atmel® ATA6670 and the INH2 pin is not used as described above, there is no possibility to use software to check if the corresponding LIN transceiver is really in Sleep mode after the “Go to sleep” command or not.

The only possibility to be sure that each of the LIN transceivers inside the Atmel ATA6670 is really in Sleep mode after the corresponding “Go to sleep” command is to switch the LIN transceivers cyclically in Normal mode for a very short period of time (minimum pulse width at the corresponding EN input is 25µs) and after that to send the “Go to sleep” command again. The ratio and frequency of switching the device to Normal mode and back to Sleep mode have a major impact on the overall current consumption for Sleep mode. With an appropriate time between these commands an acceptable current consumption in this current saving mode can be achieved.

This solution is possible if the supply pins VS1 and VS2 are interconnected as well as the two LIN transceivers are supplied separately.

3. Wake-up Source Recognition (Only Transceiver2)

The LIN transceiver 2 can distinguish between a local wake-up request (WAKE2 pin) and a remote wake-up request (LIN 2 bus). The wake-up source can be read on the TXD2 pin and the RXD2 pin in Fail-safe mode. If an external pull-up resistor has been connected between the pin TXD2 and the power supply of the microcontroller or the internal pull-up resistor of the microcontroller's TXD pin is activated, a high level at the TXD2 pin indicates a remote wake-up request (weak pull-down at pin TXD2) and a low level indicates a local wake-up request (strong pull-down at pin TXD2).

The wake-up request flag (signaled at the RXD2 pin) as well as the wake-up source flag (signaled at the TXD2 pin) are reset immediately if the microcontroller sets the EN2 pin to high.

With the help of the pins TXD2 and RXD2 the microcontroller is able to determine the wake-up reason and therefore can perform the right action for this wake-up.

The following basic C code routine shows how to carry out the procedures described above.

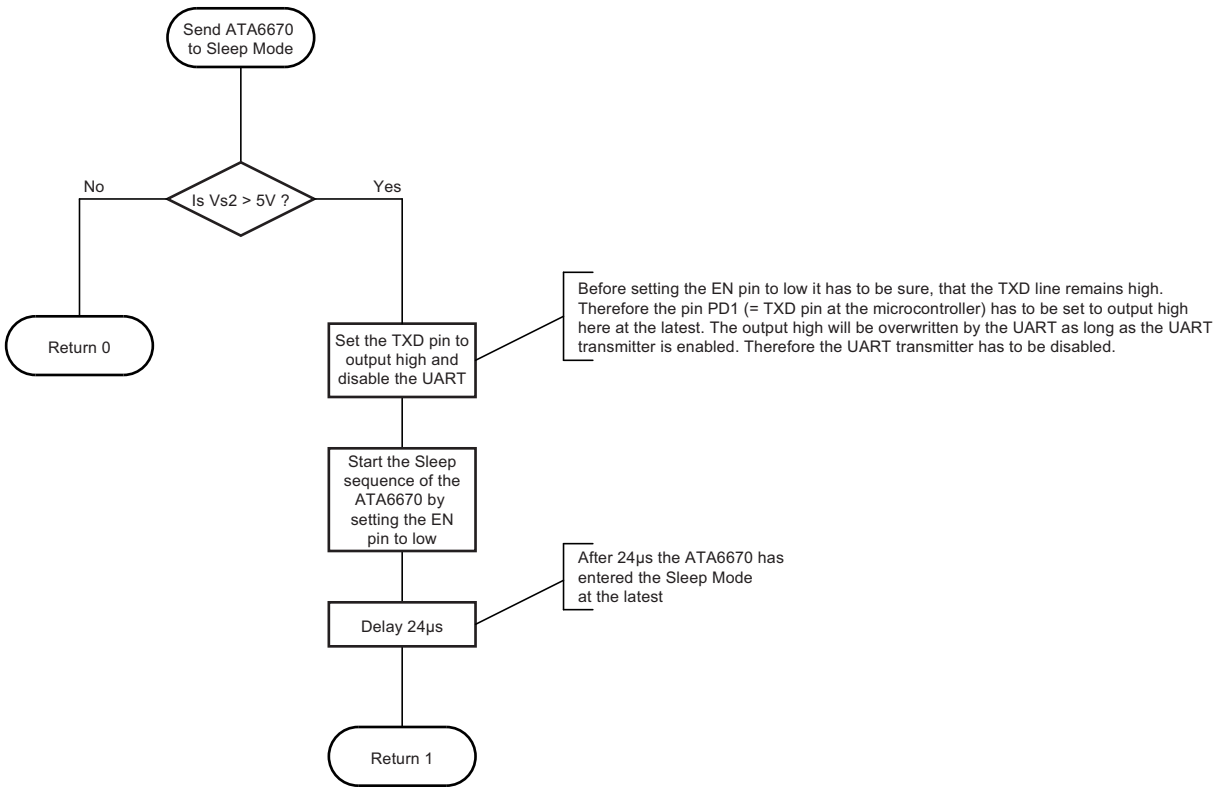
```
unsigned char Get_Wakeup_Source(void)
{
// RXD2 and TXD2 has to be configured as input pins with a pull-up
// resistor
if(TXD_INPUT_PORT & (1<<TXDPIN)) // Check the status of the TXD2 pin
       return LIN;               // If high - Remote wake-up (from
                                 // LIN bus)
else
       return WAKE;              // If low - Local wake-up
                                 // (Transceiver 2 only)
}
```

4. Program Flow

The following programming flowchart shows a secure way of switching the LIN transceivers inside the Atmel® ATA6670 into Sleep mode and recognizing the undesirable situations described here, in which the microcontroller does not recognize the real operation mode.

Using this flowchart will cause the calling function to be reported as the reason why the LIN transceiver has left the Sleep mode and why the microcontroller has remained active. The calling function then has to decide how to proceed.

Figure 4-1. Flowchart



5. Revision History

Please note that the following page numbers referred to in this section refer to the specific revision mentioned, not to this document.

Revision No.	History
9235C-AUTO-06/15	• Put document in the latest template

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**