Automatic Baudrate Detection in CANopen Networks

U. Koppe, MicroControl GmbH & Co. KG

With Layer Setting Services (LSS) it is possible to change the baudrate in CANopen networks. However, this mechanism fails in certain situations. This paper discusses an approach for an automatic baudrate detection in CANopen networks. As introduction the possible solutions to detect an unknown baudrate for CAN controllers (Software / Hardware) are presented. The paper will focus on situations where an automatic baudrate detection can fail (no traffic on the bus, error frames) and how to avoid these dead-locks.

Initial Situation

CANopen devices which do not have a hardware interface (e.g. DIP-switches, RS-232) to setup the CAN bitrate and the node address use the Layer Setting Services (LSS) protocol for that purpose. LSS is described in the CiA standard DSP-305: Two identifiers are used for communication between the LSS master (COB-ID $7E5_h$) and the LSS slave (COB-ID $7E4_h$). In order to identify a certain CANopen node, the LSS address of that node is used. The LSS address consists of four 32-bit keys (vendor-, product-, revision- and serial-number), so that every CANopen device has an unique 128-bit key for identification. By means of the “Configure Bit Timing Parameters” service the LSS Master sets the new bit timing on a LSS Slave. For CANopen devices, the following bit timings are defined:

Baudrate	*Quanta per bit*	*Sample point*
1000 kBit/s	8	75,0%
800 kBit/s	10	80,0%
500 kBit/s	16	87,5%
250 kBit/s	16	87,5%
125 kBit/s	16	87,5%
50 kBit/s	16	87,5%
20 kBit/s	16	87,5%
10 kBit/s	16	87,5%

Table 1: CANopen bitrates

Configuring a wrong bitrate for a CAN device is a critical issue. Depending on the data stream a Stuff-, CRC-, Form- or Acknowledgement-Error will occur and the CAN station will signal this by transmitting an error frame. The process of error detection and signalling stops when the device with the incorrect bitrate goes into bus-off state.

A mis-configured CANopen device that supports bitrate setup only via LSS is in a dead-lock situation now: it can not be reached via the network. The mis-configured device must be removed from the network and the stored bitrate must be evaluated by “try-and-error”, using a point-to-point connection between a LSS master and the LSS slave.

Object of this paper

The object of this paper is to present a solution for a trouble-free operation of LSS-only devices and to avoid dead-lock situations. The demands are:

- LSS-only devices can be configured within a network (no point-to-point connection necessary)
- A bus-off situation is handled autonomously
- LSS-only devices can run with CANopen modules that do not support LSS
- All defined CANopen bitrates (table 1) must be supported
- Error frames, caused by wrong bitrates, shall be avoided
- No specific identifier or data field for the CAN message required

In order to avoid the described dead-lock situation two solutions seem reasonable:

- Usage of a default fallback bitrate
- Usage of automatic baudrate detection

Default Fallback Bitrate

The “Default Fallback Bitrate” way can be implemented on any CAN controller by software: after detecting the bus-off state the device resumes with a low default bitrate (e.g. 20 kBit/s). This simple solution has several drawbacks, however. First, all devices in the CANopen network have to go through the bus-off state and they all have to be configured for the correct baudrate afterwards. Second, devices that do not support the “Default Fallback Bitrate” mechanism, will persist in the bus-off state. And finally, even if the reason for the bus-off condition was not raised by a mis-configured LSS device, the network will be forced to a “default” bitrate. As this is not acceptable, this solution is not suitable and we have a closer look to automatic baudrate detection.

Automatic Baudrate Detection

In order to meet the requirement that a running CAN network is not disturbed by error frames, it is not suitable to “test” the reaction of the network by sending out messages on all possible baudrates. The method for baudrate detection must be non-reactive for the complete CAN network. The two possible methods are:

- Measure the time of a single bit
- Suppress transmission of error frames (“Listen-Only” Mode)

Measuring the bittime

Measuring the time of a single bit does not require a CAN controller, but a good and fast timer module inside a microcontroller. The concept has already been described in the application note AP2995 from Infineon. During the detection phase the CAN controller is switched off. The timer module measures in a consecutive process the length of a single bit inside the data stream. The main problem of the method is to find one single dominant bit – and only one – inside the frame. There is always one dominant bit inside a CAN frame which is surrounded by two recessive bits: the acknowledgement bit. But this is only true if there are at least two CAN nodes on the bus. Assuming only one active node (e.g. the CANopen manager is scanning the bus), this method requires a specified identifier field or data field, otherwise it will fail. Also the method requires a timer module, whereas the clock frequency should be higher than 10 MHz. For a clock frequency of 10 MHz the difference between 800 kBit/s and 1 MBit/s is just 2 timer ticks (refer to table 2).

f_{Timer} [MHz]	***CAN bitrate [kBit/s]***	***Expected timer ticks***
10	500	20
10	800	12
10	1000	10
24	500	48
24	800	30
24	1000	24

Table 2: Timer ticks for measuring a single dominant bit

Due to the drawback that at least two active nodes are required and the high demands on the timer module this approach does not seem suitable.

Analyzing in “Listen-Only” Mode

The “Listen-Only” mode is a feature that has already been implemented in several CAN controllers, e.g. the SJA1000 (Philips), TwinCAN (Infineon) or inside the CANary family (Atmel). In “Listen-Only” mode the CAN controller can only listen to the CAN receive line without acknowledging the received messages on the bus. It cannot send any messages in this mode. However, the error flags are updated so that the bit timing can be adjusted until no error occurs.

The necessary software algorithm is

shown in figure 1. The CAN controller is initialized for acceptance of all messages (i.e. the global / local mask is set to 0). The bit timing values of the first possible CANopen baudrate (10 kBit/s) is loaded and the controller is switched into “Listen-Only” mode. Assuming that there is traffic on the network and the baudrate is correct, the message is accepted. The error registers will not change and the flag for message reception is set inside the CAN controller. This means the correct baudrate has been detected.

Assuming the baudrate is not correct, the error flags will be updated (Stuff-, CRC- or Form-Error). In this case the CAN controller is switched off and the next possible bit timing values are loaded from the baudrate table.

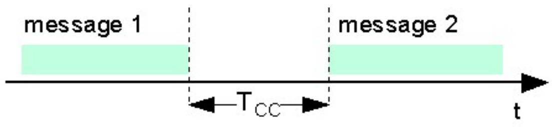

Fig. 2: Delay between CAN messages

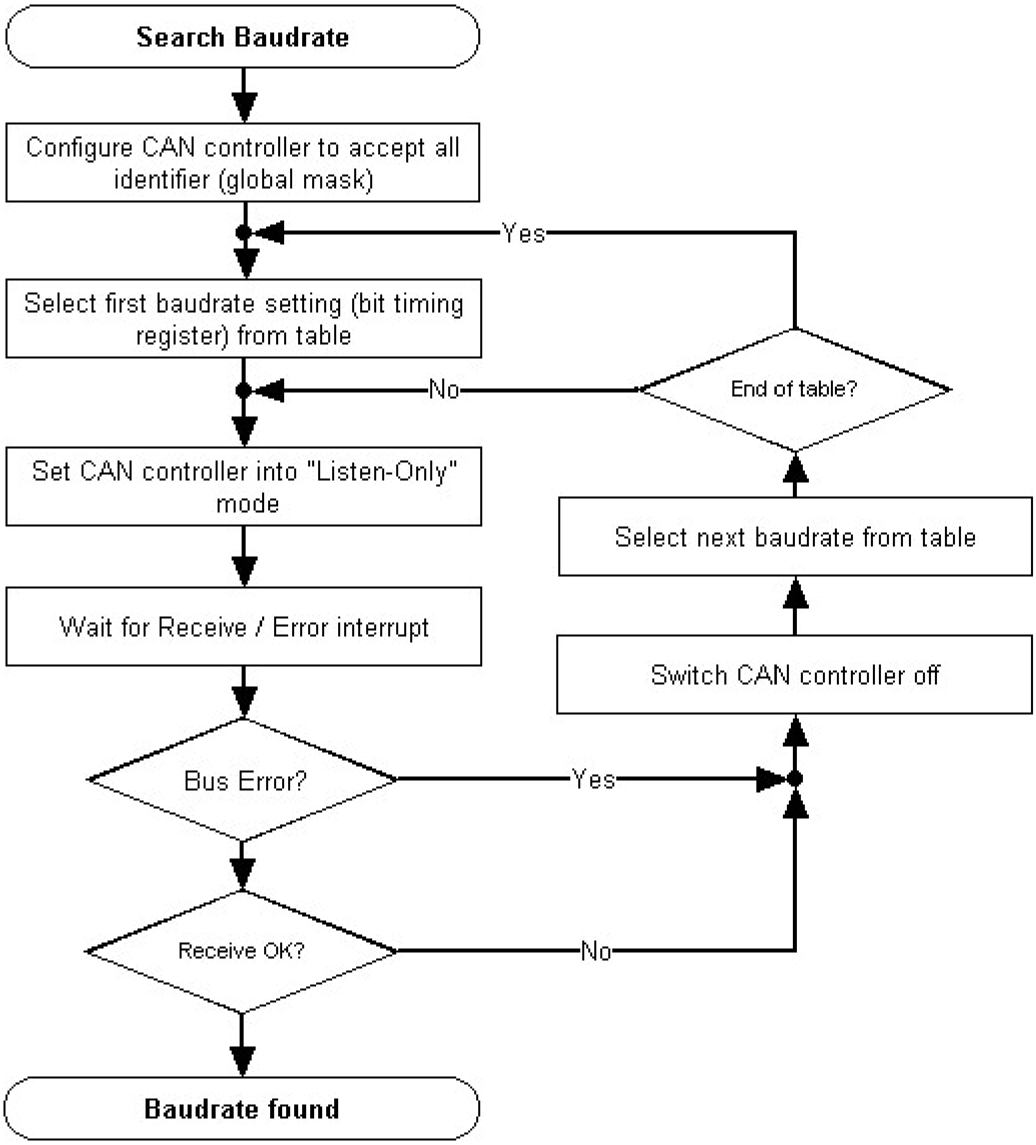

Fig. 1: Algorithm for baudrate detection in "Listen-Only" mode

The maximum number of required messages to find the correct baudrate is eight (i.e. the number of defined CANopen bitrates). However, it must be taken into account that a delay between the CAN messages is required, as shown in figure 2. The delay T_{cc} is the time to configure the CAN controller (switch CAN controller off, select next baudrate, set CAN controller into “Listen-Only” mode).

CAN controller without “Listen-Only”

CAN controller without a builtin “Listen-Only” mode require an additional external hardware. The external hardware is shown in figure 2 and consists of an AND-Gate and an OR-Gate.

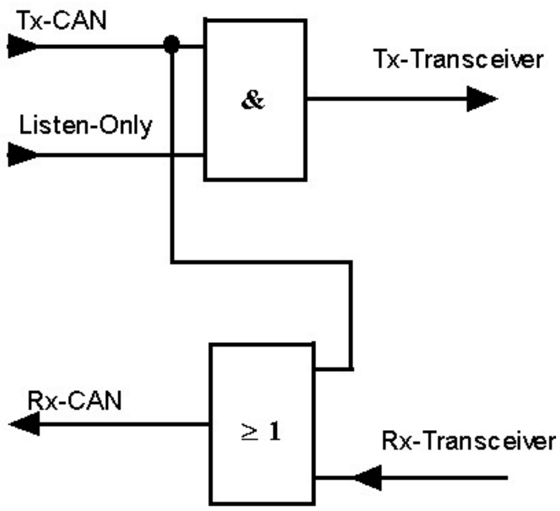

Fig. 3: External Hardware

For the given circuit it is assumend that the dominant bit is transmitted as a high level between the CAN controller and the CAN transceiver, a recessive bit is transmitted as low level. Most CAN controllers use this setup as default. CAN controller with a configurable output stage (e.g. AN82527) must be configured in that way.

An additional output pin of the microcontroller is used to select between normal operation and “Listen-Only” mode. In “Listen-Only” mode the corresponding pin of the AND-gate is held on low level. The output of the AND-gate is tied to a low level, meaning that a recessive level is at the TX-input of the CAN transceiver all the time. But the CAN controller must detect its own frames (Acknowledgement, error frames) in order to handle the internal state machine correctly. This is achieved by feeding the Tx-CAN line into an OR-gate. The Tx-CAN line is OR-ed with the receive signal of the CAN transceiver. The result is shown in table 3.

Tx-CAN	***Rx-Transceiver***	***Rx-CAN***
Recessive	Recessive	Recessive
Recessive	Dominant	Dominant
Dominant	Recessive	Dominant
Dominant	Dominant	Dominant

Table 3: Rx-CAN with additional hardware

By means of the OR-gate the CAN controller can detect the frames on the bus and its own responses (e.g. Acknowledgement).

The circuit is switched in normal operation when the “Listen-Only” pin is held on a high level. For detection of the unknown baudrate the algorithm from figure 1 is used.

Bus Traffic Requirements

In order to detect the unknown baudrate, at least one “good” reception is required. The number of necessary frames on the bus depends on the configuration of the CANopen network, as shown in figure 4.

Configuration 1 consists of a CANopen manager and devices with automatic baudrate detection. When the CANopen manager starts to configure the network, none of the CANopen slaves can sent an acknowledge. This means the very first message is repeated until the devices have found the correct baudrate.

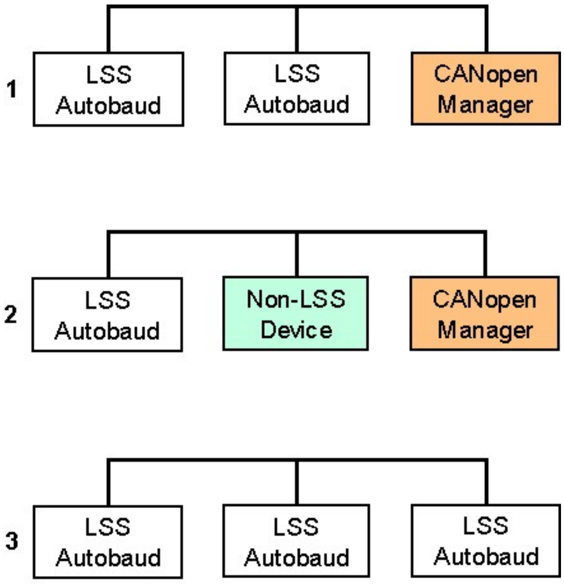

Fig. 4: Possible network configurations

Configuration 2 consists of a CANopen manager and devices that support LSS as well as devices that do not have LSS at all (hardware setup of baudrate). In that case the scanning algorithm of the CANopen manager may fail, depending on the bus traffic. As we have already seen, at least 8 CAN messages are required to find the correct baudrate. The LSS Autobaud devices will send the Boot-up message after the correct baudrate has been detected. This is the signal for the CANopen manager to configure the LSS Autobaud devices.

Configuration 3 consists of CANopen slave devices with LSS Autobaud detection only, whereas one device is a “Mini-Master”. The “Mini-Master” functionality is configured via the object 1F80h (NMT startup) in the object dictionary. By this object a CANopen device can enter the Operational state autonomously and it can start other nodes with the NMT start command. A CANopen device that is configured in that way may not use automatic baudrate detection, because the network will not start at all.

Impact of Error Frames

The automatic baudrate detection as shown in figure 1 expects correct message reception. If the CAN message is corrupted by any reason, an error frame is sent. This behaviour has an influence on the algorithm in figure 1. Suppose that the CAN controller is configured on the correct baudrate and waiting for a message. But the incoming message is corrupted, causing an error frame. Then the algorithm will select the next possible baudrate from the table, thus expanding the required time to find the correct baudrate.

Conclusion

The presented method for automatic baudrate detection in CANopen networks makes the integration of devices without a hardware interface quite simple. As a result an additional entry to the baudrate table inside the CANopen specification DSP-305 should be added, that denotes “Automatic Dectection” for an LSS device (see table 3).

Baudrate	***Table Index***
1000 kBit/s	0
800 kBit/s	1
500 kBit/s	2
250 kBit/s	3
125 kBit/s	4
reserved	5
50 kBit/s	6
20 kBit/s	7
10 kBit/s	8
Autobaud	9

Table 3: Revised CiA bit timing table

Practical tests in automotive test stands have shown that devices with automatic baudrate detection are easier to handle than those where the baudrate setup has to made via the LSS protocol. Manufactures of CANopen devices with LSS support should provide the automatic baudrate detection feature, since a better handling of those devices is achieved.

U. Koppe
MicroControl GmbH & Co. KG
Lindlaustr. 2c / D-53842 Troisdorf
+49-(0)2241 - 25 65 9 - 0
+49-(0)2241 – 25 65 9 – 11
koppe@microcontrol.net
www.microcontrol.net