AVR292: LIN "Break-in-Data" Feature of LIN/UART Controller

8-bit AVR® Microcontroller

Application Note

The LIN (Local Interconnect Network) is a serial communication protocol which efficiently supports the control of mechatronics nodes in distributed automotive applications. A LIN/UART Controller is available on some AVR® microcontrollers such as ATmega32/64/M1/C1 or ATtiny167.

Atmel LIN/UART Controller

- **Hardware Implementation of LIN 2.x (LIN 1.x Compatibility)**
- **Small, CPU Efficient and Independent Master/Slave Routines Based on “LIN Work Flow Concept” of LIN 2.x Specification**
- **Automatic LIN Header Handling and Filtering of Irrelevant LIN Frames**
- **Automatic LIN Response Handling**
- **Extended LIN Error Detection and Signaling**
- **Hardware Frame Time-out Detection**
- **“Break-in-data” Support Capability**
- **Automatic Re-synchronization to Ensure Proper Frame Integrity**
- **Fully Flexible Extended Frames Support Capabilities**

1. Introduction

This document describes the behavior of the LIN/UART Controller when it detects an unanticipated BREAK field during an otherwise normal LIN transfer. This event is what we refer to as "Break-in-Data".

The protocol says:

> “*A slave task shall always be able to detect the break/sync field sequence, even if it expects a byte field (assuming the byte fields are separated from each other). A desired, but not required, feature is to detect the BREAK/SYNC field sequence even if the break is partially superimposed with a data byte. When a BREAK/SYNC field sequence happens, the transfer in progress shall be aborted and processing of the new frame shall commence.*”

Reference: LIN Protocol Specification, Revision 2.1, November 24 2006, Page 29, § 2.3.1.2 Sync byte field.

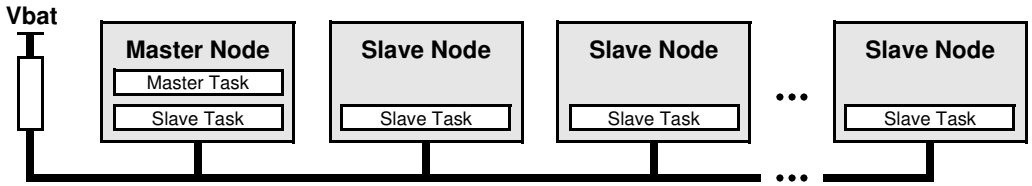
8125A–AVR–03/08

2. BREAK Field & LIN Header

2.1 Master & Slave Nodes

A LIN cluster consists of one master task and several slave tasks. A node is a single connection to the LIN bus. A master node contains the master task and a slave task. All other slave nodes contain a slave task only. The master task decides when and which frame shall be transferred on the bus. The slave tasks provide the data transported by each frame.

Figure 2-1. LIN Cluster



2.2 Typical LIN Transfer

The header consists of a BREAK and SYNC fields followed by a PROTECTED IDENTIFIER field. The identifier uniquely defines the purpose of the frame. The slave task appointed for providing the response associated with the identifier transmits it. The response consists of a DATA field and a CHECKSUM field. The slave tasks waiting for the data associated with the identifier receives the response and uses the data transported.

Figure 2-2. Master and Slave Tasks Behavior in LIN Frame

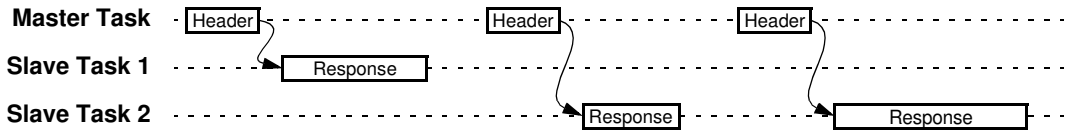
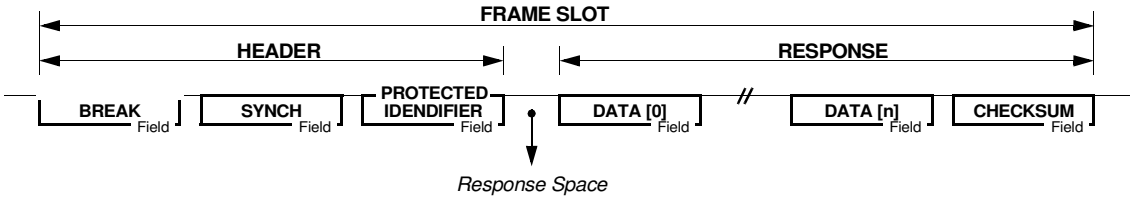


Figure 2-3. Structure of a LIN Frame



2.3 Disruption of Sequence

The master task (in the master node) transmits headers based on a schedule table. The schedule table specifies the frames and the interval between the start of a frame and the start of the following frame.

Usually, according to the schedule table, the master task allows enough time for the slave tasks to handle the response.

But sometimes, the timing defined by the schedule table is not respected. The reasons for the disruption can be:

- A change of schedule table,
- A priority frame to handle,
- Another unanticipated event,
- ...

In any case, the LIN/UART Controller has to consider any form of errors in schedule scheme.

3. "Break-in-Data"

The consequence of a disruption of sequence must be studied for both slave and master nodes during the following transfer states:

- Header sequence,
- Response space,
- Response sequence.

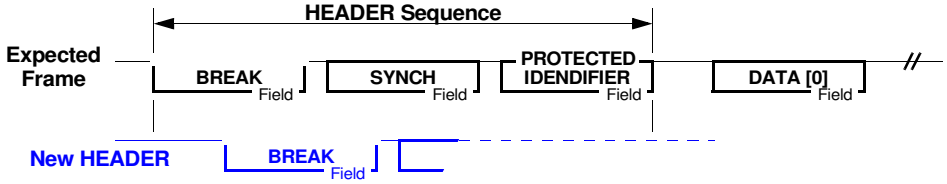
The following paragraphs describe all the cases of disruption of sequence and how the LIN/UART Controller recovers from them.

3.1 Master Node

A master node executes master and slave tasks.

3.1.1 Header Sequence

Figure 3-1. BREAK Field During Header Sequence in a Master Node



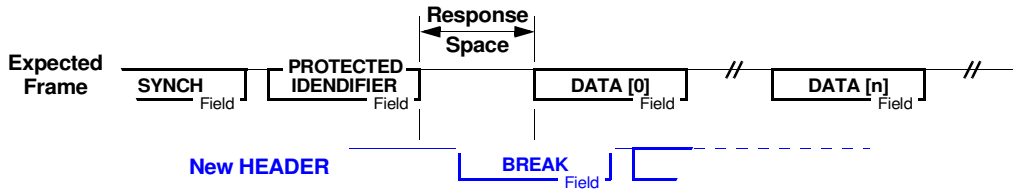
In the master node, the LIN/UART Controller is responsible for the transmission of the header. To perform this, the master task invokes the ‘Tx header’ command by writing to LINCR register. Automatically, the LIN/UART Controller sends the three fields of the header without further user intervention.

During this phase of transmission, a new ‘Tx header’ command could be initiated. This command will not be treated because one of the standard features of the LIN/UART Controller is to lock the command register access while the controller is busy (transmitting).
The only way to re-start the transmission of a (new) header is to enter first a ‘LIN abort’ command and subsequently the ‘Tx header’ command. This way of operating is recommended in a LIN software driver (c.f. AVR286 application note).

Initiating the ‘LIN abort’ command while a header is being transmitted (LBUSY flag set) sets the LABORT error flag. The application is thus informed of the disruption of sequence.

3.1.2 Response Space

Figure 3-2. BREAK Field During Response Space in a Master Node



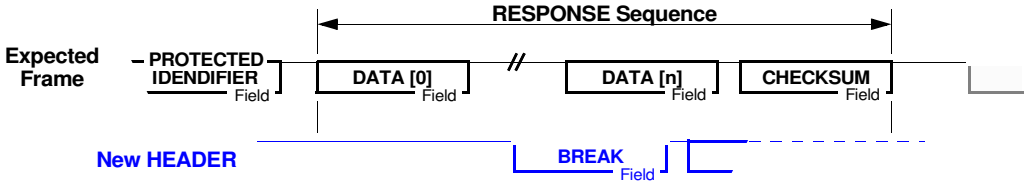
The LIN/UART Controller is not busy, having terminated the previous header generation.

If the master task now has to send a (new) header, this permitted operation will not set an error flag. However the slave task in the master node may be in the process of replying data to the previous header though it has not yet written to the LINCR register. As long as the slave task checks the LBUSY flag (set once again by the new header transmission) before writing the ‘Rx response’ or ‘Tx response’ command and in the case that this flag is set aborts the reply operation, then there is no conflict between master and slave tasks.

Respecting this procedure ensures the LIN operation integrity.

3.1.3 Response Sequence

Figure 3-3. BREAK Field During Response Sequence in a Master Node



Again, it is the same LIN/UART Controller which is responsible for the slave and the master tasks. To perform this, the slave task invokes the ‘Rx response’ or the ‘Tx response’ command by writing to LINCR register. Automatically, the LIN/UART Controller sends the response section of the LIN frame (data + checksum) without further user intervention.

During this phase, a ‘Tx header’ command could be initiated. This command will not be treated because the LIN/UART Controller locks the command register access while the controller is busy (receiving or transmitting the response).
The only way to start the transmission of a (new) header is to enter first a ‘LIN abort’ command and then the ‘Tx header’ command. This way of operating is recommended in a LIN software driver (c.f. AVR286 application note).

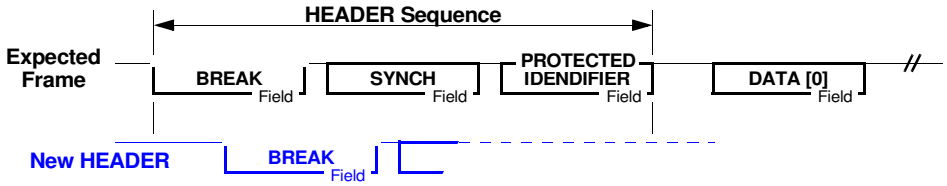
Applying the ‘LIN abort’ command while the LIN/UART Controller receives/sends a response (LBUSY flag set) sets the LABORT error flag. The application is thus informed of the disruption of sequence. By reading LTXDL[3..0] and LTXDL[3..0] in LINDLR register the application is informed of how many data bytes were successfully transferred before the data transfer was aborted.

3.2 Slave Node

A slave node only executes slave tasks. Once the LIN/UART Controller is enabled, it is in continuous detection of the BREAK field, whatever the task in progress. It is a background hardware task of the LIN/UART Controller.

3.2.1 Header Sequence

Figure 3-4. BREAK Field During Header Sequence in a Slave Node



If the LIN/UART Controller detects a dominant level in the range 9.5-28 bits (tolerance of re-synchronization), this is recognized as a BREAK field and the SYNC field detection and its measurement will start.

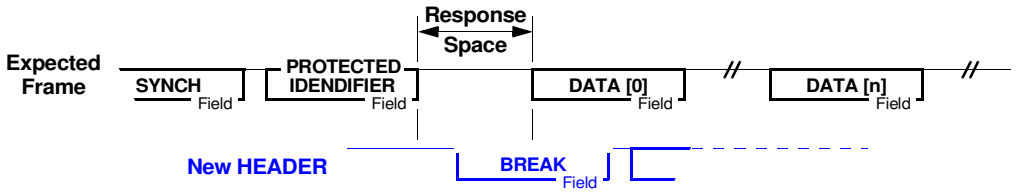
Possible cases:

1. The new BREAK field overlaps exactly on the old one (same length):
   The acquisition of the header continues as if nothing has happened and the processing of the frame can follow.
2. The new BREAK field and the old one produce a longer resultant field (logical AND):
   If the frequency is stable and tuned to the nominal value, the length of the resultant BREAK will have at the most twice the standard value (26 bits versus 13 bits). This is in accordance with the LIN protocol BREAK field tolerance (10-28 bits).
   The acquisition of the header continues absorbing the disturbance and the processing of the frame can follow.
   The new BREAK field is inserted during the SYNC field (logical AND):
   The (old) header acquisition stops. Either a synchronization error flag (LSERR) or a framing error flag (LFERR) is set and informs the application of the disruption of sequence. A framing error is detected only if the timing slot reserved for the stop-bit is forced to a dominant level by the new BREAK field.
   The processing of the new frame can start.
3. The new BREAK field is inserted during the PID field (logical AND):
   The (old) header acquisition stops. Either a bit error flag (LBERR) or a framing error flag (LFERR) is set and informs the application of the disruption of sequence. A framing error is detected if only the slot reserved for the stop-bit is forced to a dominant level by the new BREAK field.
   The processing of the new frame can start.

Thus the LIN/UART Controller with the help of error flags allows the slave tasks of the slave nodes to handle successfully all these possible conditions.

3.2.2 Response Space

Figure 3-5. BREAK Field During Response Space in a Slave Node



During this phase, the slave task must check the in-coming LIN ID and must initiate a reception or a transmission of a response if the node is concerned with the received LIN ID.

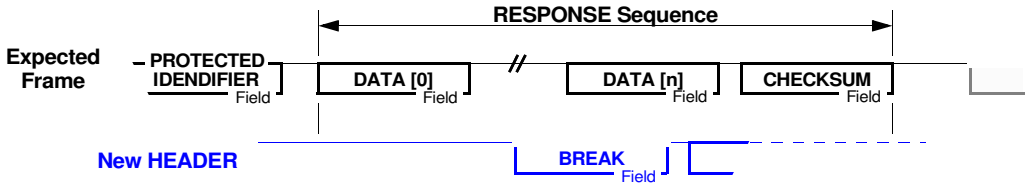
During this phase, the LIN/UART Controller automatically anticipates a byte reception even though it has not yet been ordered to take a specific action (‘Rx response’ or ‘Tx response’) by the slave task.

Possible cases:

1. 9 bits of the BREAK field have been already received:
   9 bits of dominant level will suffice to switch the LIN/UART Controller into anticipating a tenth stop bit for a normal byte reception. Due to the fact that it is a BREAK field that is being received, the expected stop bit will be absent. The LIN/UART Controller will detect this situation and will signal a framing error by setting LFERR flag. This informs the application of the disruption of sequence.
   The processing of the new frame can start.
2. The node has just issued the ‘Rx response’ command:
   Due to the anticipated byte reception, the BREAK field has been partially received (< 9 bits). Reception of this BREAK field continues until the tenth bit has been received. Again, the absence of the expected stop bit will be signalled and a framing error flag (LFERR) is set. This informs the application of the disruption of sequence.
   The processing of the new frame can start.
3. The node has just issued the ‘Tx response’ command:
   In this case, the attempt at transmission is aborted because, before sending a start bit, the LIN/UART Controller verifies that the LIN bus is free (recessive level). Should the LIN bus not be free the LIN/UART Controller will signal this fact and will set the bit error flag (LBERR). This informs the application of the disruption of sequence.
   The processing of the new frame can start.

3.2.3 Response Sequence

Figure 3-6. BREAK Field During Response Sequence in a Slave Node



The term "Break-in-Data" fits most exactly with this situation, a BREAK field appears during a response (reception or transmission).

1. The node is transmitting the response:
   Due to the length and the level of the BREAK field, either a bit error flag (LBERR) or at least a framing error flag (LFERR) will be set to inform the application of the disruption of sequence.
   The processing of the new frame can start.
2. The node is receiving the response:
   Due to the length and the level of the BREAK field, a framing error flag (LFERR) will be set to inform the application of the disruption of sequence.
   The processing of the new frame can start.

4. Conclusion

In every case, a BREAK field aborts the on-going operation and a new LIN acquisition sequence starts. The frame that aborts a previous one predominates and is not lost. The application is always informed of such disruptions of sequence by the setting of LIN error flags in the LIN/UART Controller.

“Break-in-Data” is fully supported in the Atmel LIN/UART Controller.

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature