Bulletin of the *Transilvania* University of Braşov
Series I: Engineering Sciences • Vol. 6 (55) No. 2 - 2013

SOLUTIONS TO IMPROVE THE FEATURES OF LOCAL INTERCONNECTED NETWORKS

A.C. STANCA[1] **F. SANDU**[1]

Abstract: *The Local Interconnected Network (LIN) is cheap - being short and low-speed - but very practical for many appliances. Its architecture has a single Master and multiple Slaves for connecting sensors and actuators in automotive applications. The use of LIN can be extended to other industrial applications having similar non-critical requirements - as the automotive systems do. This paper is proposing solutions to improve some features of LIN, in order to extend this applicative area. The characteristics improved are the cable length and the number of network nodes, by three solutions developed and presented in detail.*

Key words: *LIN, repeater, bridge, microcontroller, transceiver.*

1. Introduction

Cost-effectiveness has driven LIN design, so its main characteristics are [5]:

- bidirectional network (transmit-receive) on a *single* wire (except the power-wires);
- cable length is *limited* to 40 m;
- transmission speed is self-controlled but the transfer rate is *limited* to 20 kbps;
- microcontrollers use *very cheap* RC oscillators;
- number of nodes is *limited* to 16.

In a LIN, there are Slave nodes which are queried to deliver data in the network, called P-type nodes (Producers).

The other Slave nodes are not queried but they are benefiting from data returned by Producers, called C-type nodes (Consumers).

Even with these limited performance, LIN can successfully replace fault-tolerant CAN (Controller Area Networks) and one-wire CAN [4] in automotive appliances.

2. Objectives

The objectives of this paper are to explore the possibilities:

- to extend the cable length for LIN beyond the 40 m and also;
- to increase the number of nodes for LIN.

Both improvements should be achieved without major increase of LIN costs.

Our solution to improve the above-mentioned characteristics is the development and use of LIN repeaters and bridges.

3. Materials and Methods

We have chosen for our implementation, as core for the repeaters [1], the PIC12F1822 microcontroller; this is a small and cheap device (thus preserving the low cost of our approach) having the following main characteristics [7]:

- flexible clock structure with a

---

[1] Electronics and Computers Dept., *Transilvania* University of Braşov.

precision 32 MHz (±1%) internal oscillator block;

- an EUSART compatible with RS-232, RS-485 & LIN and having an Auto Baud Rate Detect mode; this feature also recommends the device for our purpose;
- operating voltage range: 1.8-5.5 V, a feature that allows the network to work in harsh conditions.

For the bridge we have used the same type of microcontroller; we implemented a special software solution, described in [1], to get a serial communication equivalent port - another contribution to cost-effectiveness.

As interface between the microcontroller and the LIN Bus it was used the LIN J2602 Transceiver MCP 2003 [6], with the following main features:

- compliance with LIN Bus Specifications 1.3, 2.0 and 2.1;
- support for LIN Baud Rates with LIN-output compatible driver;
- wide supply voltage range, 6-27 V DC;
- standard USART and EUSART interfacing (appropriate for PIC μC, as shown above).

The specific software integration and functional testing was accomplished by the authors.

3.1. Repeater LIN.Co

To extend cable length beyond the standard limits as well as the number of C-type Slave nodes, without performance decrease, we used the Repeater LIN.Co (Consumers only) shown in Figure 1. In a standard LIN they are manipulated a maximum of 64 messages (usually defined by the manufacturers). The maximum number of nodes is 16 - limited by electrical characteristics of the network. Any nets (the segments) added to the base network don't exceed the number of 16 nodes each.

Connecting the two transceivers enables message transfer from the LIN BUS 1 segment (with Master and the both types of Slave nodes) to the LIN BUS 2 segment (with only C-type Slave nodes). Transceiver LIN 1 is set for reception (Listening mode) and Transceiver LIN 2 is set for transmission (Talking mode). So, we interconnected "sub-net" segments that don't exceed the maximal number of 16 Slave nodes each. For the simpler "subscriber" nodes (according to [5]) - that read transmitted data (and can process them locally), without preparing any reply - their number can be extended in supplementary segments preceded by "unilateral" LIN.Co repeaters.

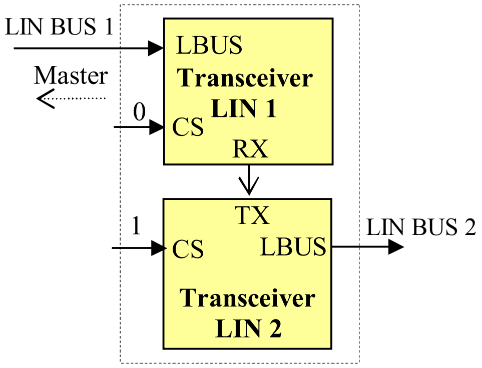

Fig. 1. *LIN.Co repeater structure*

Figure 2 shows an example of a LIN extended through the use of two LIN.Co repeaters.

3.2. Repeater LIN.PC

We implemented also a more complex repeater, including a PIC μC dedicated to data flow management - Figure 3. We called it LIN.PC (Producer & Consumer) as it extends also the number of P-nodes (besides the number of C-nodes).

When the Master node launches the query message, LIN Transceiver 1 is in the Listening mode - the CS input being set to 0 (via the CS1 signal); the LIN Transceiver 2 is in the Talking mode - his CS input is set to 1 (via CS2 signal).

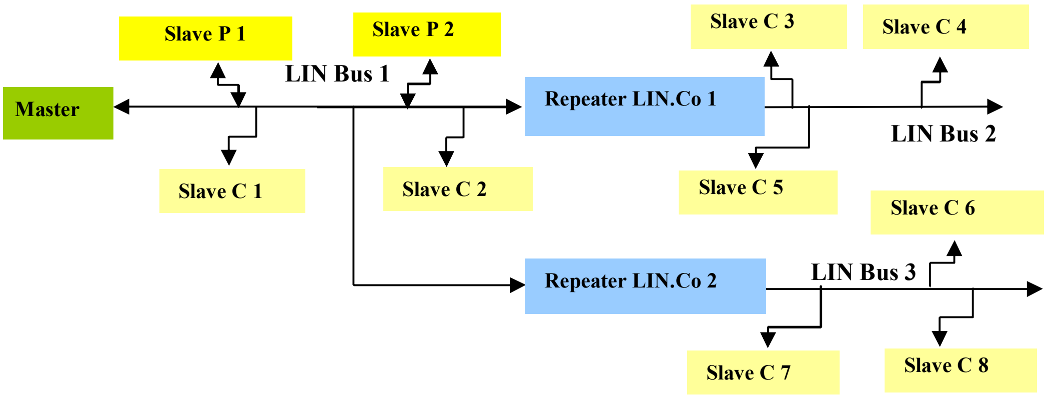

Fig. 2. *LIN extended by LIN.Co repeaters*

In this way, the message goes from the LIN BUS 1 segment to the LIN BUS 2 segment, directly without the mediation of the microcontroller. The message is also read by the microcontroller and, if the message has as destination a node of P-type that belongs to the LIN BUS 2 segment, it will switch to the complementary status at the end of the message delivered by Master node.

The CS input of Transceiver LIN 1 is set (transmission on) and the CS input of Transceiver LIN 2 is reset (transmission off). Thus, the response message returned by the identifier (ID) owner node passes from the LIN BUS 2 segment to the LIN BUS 1 segment.

The Repeater LIN.PC is actually a P-type node that is interrogated by the Master. It has a identifiers (ID-s) list recognized in the destination segment.

So, in case of interrogated Slave nodes (“publishers”, according to [5]) - the LIN.PC repeater first enables passing of messages from the main segment (that includes the Master) to the secondary segments (that extend the network) and then, LIN.PC enable the answer message to be transferred in the main segment, towards the Master.

Thus, we obtained a cable length that can be extended up to 40 m multiplied with the number of PC repeaters.

Also, it can be increased the number of P and C type nodes without modifying the standard message structure.

Figure 4 shows the example of a LIN extended through the use of LIN.PC repeaters.

The query of a Slave belonging to a segment managed by Repeater LIN.PC is done in two steps:

- in the first step, the Master interrogates the Repeater nodes that will prepare themselves to pass the next message, if it is the case;
- in the second step, the next message is passed in the segment that contains the queried node.

3.3. LIN to LIN Bridge

If two LIN with different destinations must exchange information, the structure of LIN to LIN Bridge (“Bridge L2L”) is recommended. The principle of such a LIN node is presented in Figure 5.

The Bridge we implemented doesn’t belong to the standard LIN configuration - so, there’s no LIN standard protocol sub-set dedicated to messaging through this Bridge. We had to devise a messaging mechanism described below.

In normal state, both transceivers are in the Listening mode, the signals of the inputs control CS being CS1 = 0 & CS2 = 0.

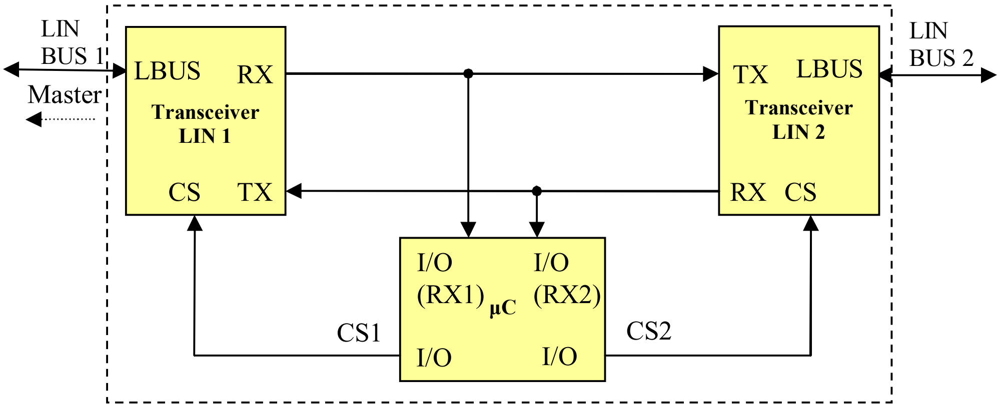

Fig. 3. *LIN.PC repeater structure*

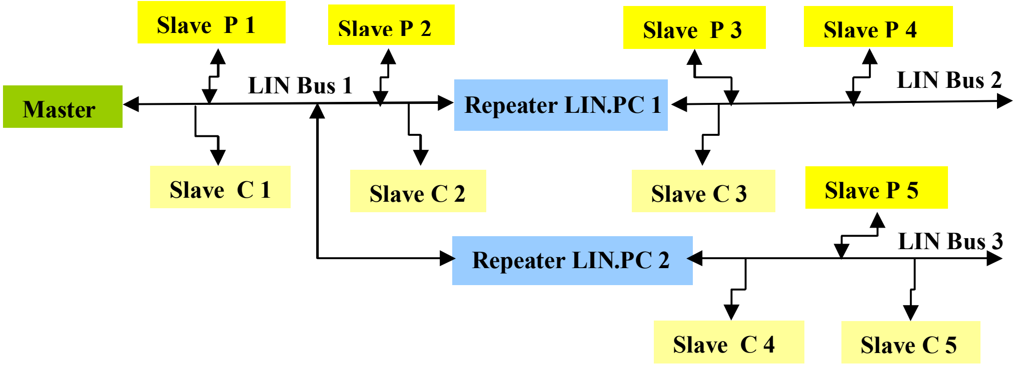

Fig. 4. *LIN extended by LIN.PC repeaters*

In case the Master 1 sends a query message in its own LIN BUS 1 segment, the microcontroller (core of the Bridge L2L) receives this message on its RX1 input. An interrupt serving routine will inspect the message identifier and will take the decision:

a) not to retransmit the message towards LIN BUS 2, if the identifier is not found in the list associated with this segment; the message already arrived at all nodes of the LIN BUS 1 segment; in this case CS1 and CS2 are maintained in 0 logic;

b) to retransmit the message towards LIN BUS 2, if the identifier is found in the list associated with this segment; in this case CS1 = 0 and CS2 = 1, which means that the Transceiver 2 will pass the signal present at its TX input to LIN BUS 2.

The sequences are similar if Master 2 sends query messages on its own segment of LIN BUS 2.

One should note that, in both cases, a) and b), the message is temporarily stored.

After filtering, the message in case a), is ignored, instead, in case b), before being transmitted on the complementary segment it is waited the header message of the Master of this segment (during this period CS1 = CS2 = 0) and the associated response of interrogated Slave.

After the arrival of the entire message from this Master, the Bridge transmits them a particular message (Busy) which inform about the next message that is the header message of the complementary Master.

During the transmission of the Busy message and of the header message of the complementary Master, the transceiver of the destination network is set in Talking mode by the appropriate CSn signal.

With this device, we extended the cable length to 80 m and the number of nodes to 32.

The speed in coupled networks can be kept close to the maximum level.

In Figure 6 there are shown two LINs using a Bridge node L2L.

Each network contains, besides the Master node, two P-type and two C-type nodes.

4. Results and Discussions

The Repeater LIN.Co was tested for a network with 10 P-type Slave nodes in a network segment and 10 C-type Slave nodes in another network segment, with a baud rate of 9600, without errors.

In Figure 7 it is presented a temporal diagram of a simple test configuration.

This test configuration is using a Master node (M), a Repeater LIN.PC (R-LIN.PC), 2 P-type (SPn) and 4 C-type (SCn.m) nodes.

The SP1 node manages the identifier 1, and the SP2 node the identifier 2. The SC1.1 and SC1.2 nodes read data transmitted from SP1 and the SC2.1 and SC2.2 reads data transmitted from SP2.

In case 1, the Master interrogates the SP1 node that belongs to the same segment. Messages from the SP1 node arrive to SC1.1 after a time t_r and to SC1.2 after a time $t_1 = t_r + t_p$, where t_r is the response time of the LIN Slave (as defined in [5]) and t_p is the propagation time of the repeater.

In case 2, the Master interrogates the SP2 node that belongs to the other segment.

The data messages from the SP2 node arrive to SC2.2 after a time $t_1 = t_r + t_p$ and to SC2.1 after a time $t_2 = t_r + 2t_p$.

The Repeater LIN.PC was tested for a configuration with 5 P-type and 5 C-type Slave nodes in a network segment and 5 P-type and 5 C-type Slave nodes in another network segment, for a rate of 9600 bps, without any malfunction.

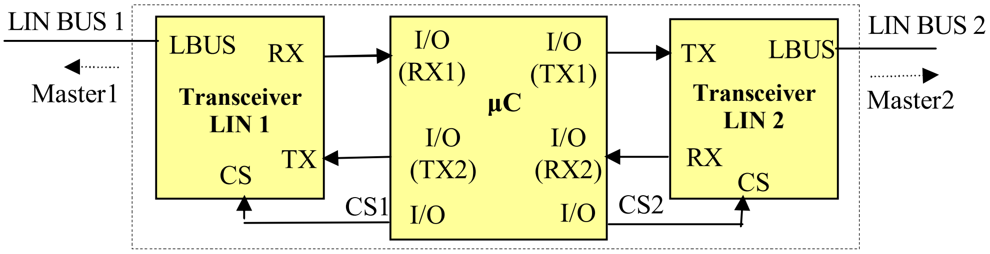

Fig. 5. *Structure of L2L Bridge*

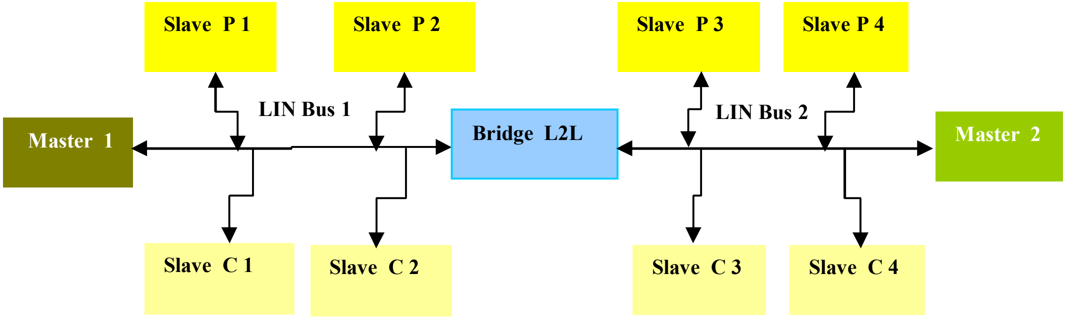

Fig. 6. *LIN extended through a Bridge L2L*

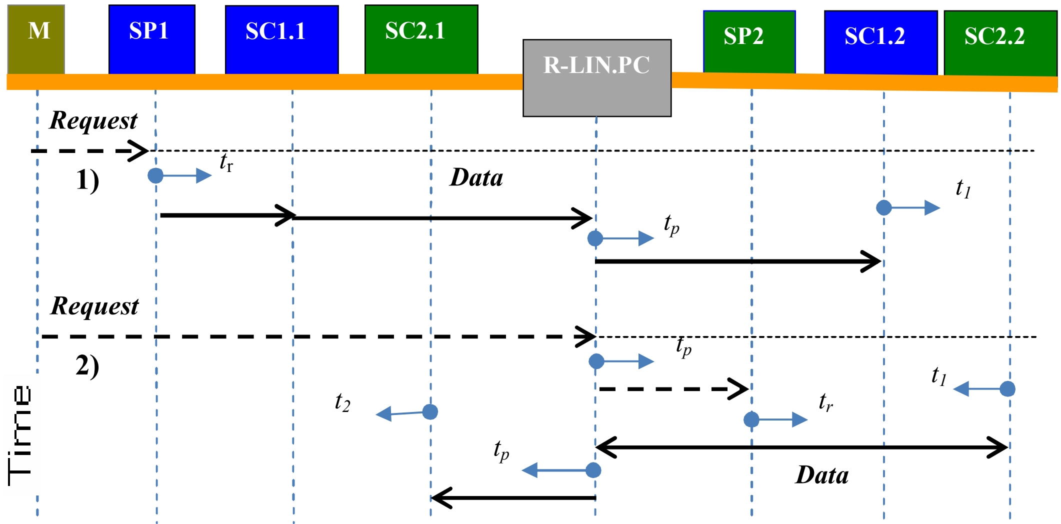

Fig. 7. *Temporal diagram of a LIN extended with a repeater LIN.PC*

For this typical transmission speed of 9600 bit/s (or “bps”), the critical parameter corresponding to the time frame of a maximal message (124 bits) equals 124 b: 9600 bps = 12.91 ms, rounded to 15 ms (as the time quantum of the network is 5 ms). Considering a time reserve of 40% the message length (as required by the standard), we get this way the minimum interrogation interval, 15 ms · 1.4 = 21 ms.

In order to assess the impact of the bus increase in length using LIN.Co, we should notice that the repeater is unilateral (it doesn’t manipulate responses).

We should take into consideration the transceiver timing, 4 µs delay for transmission and 6 µs for receiving.

So, the delay is 0.019-0.028% of the 21 ms minimal interrogation interval.

Considering also a line propagation time of 6 µs, the maximal delay of 6 + 6 = 12 µs wouldn’t impact the response time with more than 0.056% of this interrogation interval.

In case of the LIN.PC repeater, that is also able to return data, the greatest delay belong to the data fields, being twice the delay of the header, then 24 µs, representing 0.12% of the interrogation interval.

The added line segments (40 m at maximum) aren’t “chained” but “star-bus” so delays won’t be cumulated.

In Figure 8 it is presented a temporal diagram for a test configuration with two networks coupled through a Bridge L2L.

The first one is around Master 1, having one P-type Slave node, SP1, and two C-type Slave nodes, SC1.1 and SC 2.1.

The other one is around Master 2, with a P-type Slave node, SP2 and a C-type Slave node, SC 2.2.

SC1.1 reads the data transmitted from SP1, and SC2.1 & SC2.2 read the data transmitted from SP2.

In case 1, the data arrives to SC1 from SP1 with a delay equal to t_r.

In case 2, the data arrive at SC2.2 with a delay $t_1 = t_w + 2t_r + 2t_b$ and at SC2.1 with a delay $t_2 = t_1 + t_p$, where t_w is the wait request time and t_b the inter-byte time (as defined in [5]).

The LIN Bridge L2L was tested, without any problem, for a network with 2 P-type and 4 C-type Slave nodes in a network segment and 2 P-type and 4 C-type Slave nodes in another network segment, with a rate of 9600 bps.

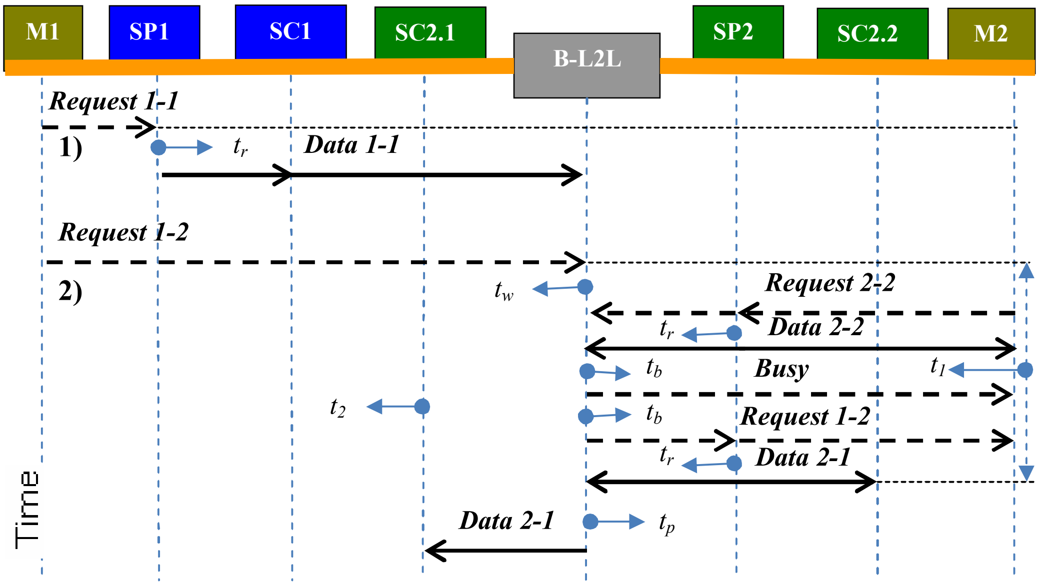

Fig. 8. *Temporal diagram for two LINs coupled through a Bridge*

The practical testing was done in a configuration for the automation of thermal control in a greenhouse.

The module from Figure 9 was designed for a distributed control system dedicated to the greenhouse micro-climate.

The control board is "learning" the temperature thresholds - the currently measured values are memorized by pressing the adequate keys.

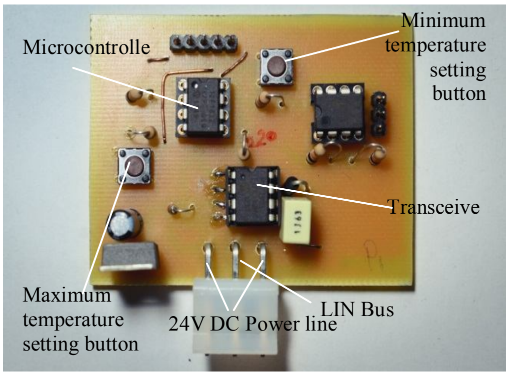

Fig. 9. *LIN.PC type node in a temperature control board*

The module from Figure 10 is a correspondent to the one presented above.

This module includes a LIN.Co node that is using data from the associated module with LIN.PC, in order to drive an actuator.

In order to increase temperature, such a LIN.Co node commands the powering of a heating element; another node of this type sends the temperature decrease commands to a cooling fan. If temperature has a normal value, between the two limits, no actuator is activated.

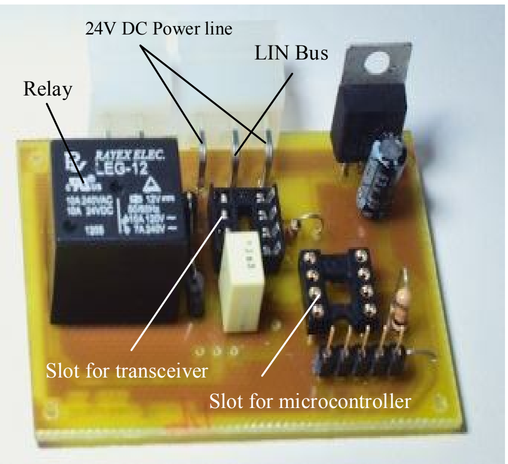

Fig. 10. *LIN.Co type node in a temperature control board*

The Bridge was tested in an experiment that interconnects two such greenhouses that use common data on external measurement - enlightenment, exterior humidity and temperature - important parameters for the micro-climate control in each of the separated greenhouses.

5. Conclusions

The implementations presented in our paper were not used yet to their upper performance limit.

The developed devices have been tested only for 20 nodes (10 of P-type and 10 of C-type) in two network segments with a rate of 9600 bps.

The results obtained were compliant with the expected ones in terms of signals integrity.

Future experiments and deployments of various industrial or domotic applications [2], [3] with LIN are expected to confirm the performance of these solutions.

References

1. Butler, D., Schmidt, T., Waclawczyk, T.: *LIN Protocol Implementation Using PICmicro MCUs*. Microchip Technology Incorporated, 2002.
2. Ogrutan, P.: *Microcontrollere și controllere grafice Fujitsu* (*Fujitsu Microcontrollers and Graphical Controllers*). Braşov. Transilvania University Publishing House, 2006.
3. Stanca, A.C., Sandu, V., Văduva, R., Nemeth, O.: *Distributed System for Indoor Temperature Control*. In: Annals of the University of Craiova, Electrical Engineering Series **36** (2012), p. 290-295.
4. Voss, W.: *A Comprehensible Guide to Controller Area Network*. Copperhill Media Corporation, 2005.
5. *LIN Specification Package*. Revision 2.2A, December 31, 2010; LIN Consortium. Available at: www.lin-subbus.org.
6. *MCP2003/4 LIN J2602 Transceiver Data Sheet*. Microchip Technology Inc., 2010.
7. *PIC12(L)F1822/PIC16(L)F1823 Data Sheet*, *8/14-Pin Flash Microcontrollers with XLP Technology*. Microchip Technology Inc., 2010-2012.