SPI-TO-LIN BRIDGE – SJA1124

Opening up New Horizons in Automotive LIN Commander Solutions; A New Step in ECU Integration and Design

LIN bus systems are mainly used in the body domain of vehicles, and the number of nodes in the car is constantly increasing. This article introduces the concept of an SPI-to-Multiple LIN bridge that will significantly change the way how LIN automotive bus systems will be connected to the microcontroller for the electronic control units (ECUs) in the car.

The SPI-to-LIN bridge is independent from the number of LIN/UART channels, hence allows the host MCU to be fully tailored to application performance requirements without constraints on the number of LIN channels. Besides significantly reducing the bill of materials (BOM) and board space consumption, this concept also greatly enhances the flexibility and scalability of multi-LIN channel architectures.

We conclude with an example of a multi-channel LIN commander application with an SPI-to-LIN bridge, based on NXP's SJA1124. The SJA1124 is a LIN transceiver incorporating four LIN commander channels, each with integrated commander protocol controller and commander termination. By integrating the commander termination along with the commander protocol controller, BOM and board space is reduced.

INTRODUCTION

More and more applications have multiple LIN commander channels. Moreover, the number of LIN channels per application is also increasing, whereas the available space for these applications is typically shrinking.

This paper describes SPI-to-LIN bridge-based solutions for multi-channel LIN applications. It explains how board space and BOM are reduced and how the scalability and flexibility of the application architecture and MCU variety are enhanced.

MCU LIMITATIONS

ON-CHIP LIN FEATURES

There is a wide range of integrated MCU features supporting LIN applications, varying from a simple UART to a full-featured LIN protocol controller. The number of integrated LIN channels also varies but typically depends on the MCU performance platform.

TABLE OF CONTENTS

PIN COUNT

For each LIN channel, at least two pins are required: TXD and RXD. In addition, one or more mode control pins might be necessary to switch the LIN transceiver mode from low power mode to normal mode and vice versa.

Thus, for applications with multiple LIN commander channels the number of necessary pins accumulates with at least a factor of two. For instance, 16+ pins are required for eight LIN channels. The example in Figure 1 illustrates the necessary MCU pin count for two quad LIN transceivers, connected in the traditional way with pairs of TXD and RXD per LIN channel.

If the MCU pin count is already critical, the high number of pins required for multiple LIN commander channels might force the selection of the next higher (and more expensive) MCU family variant with more pins, if available — which is not always the case. This leads to a significant increase of the BOM as well as cost.

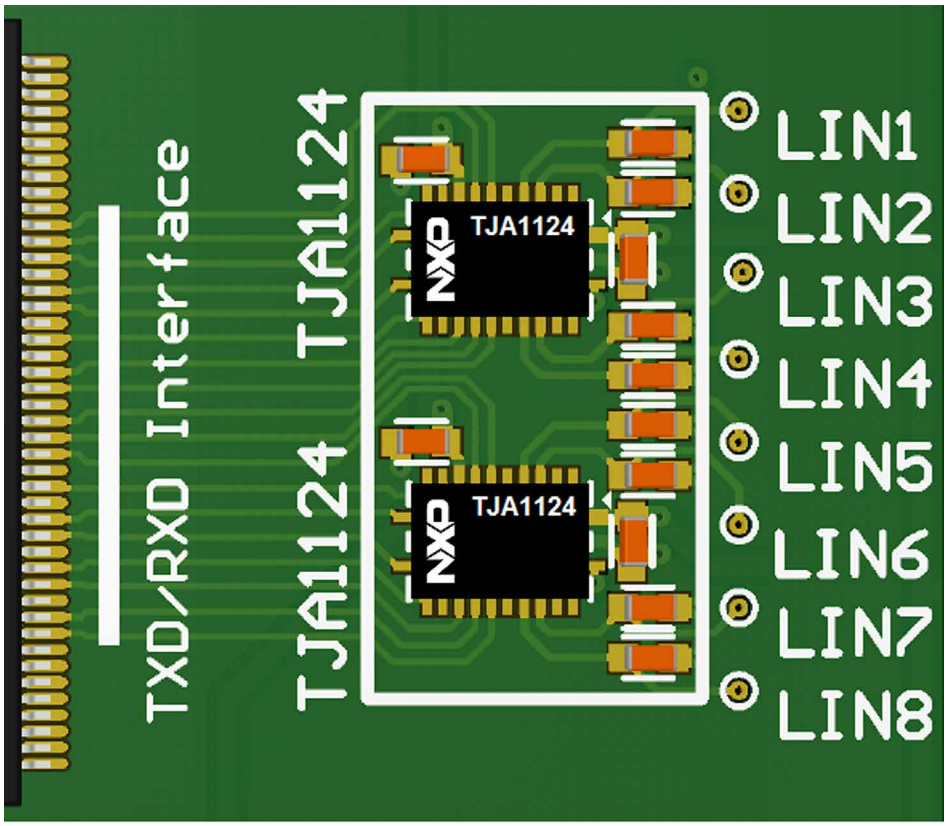

Figure 1: MCU pin count example: 18 MCU pins for eight LIN commander channels

NUMBER OF ON-CHIP UART/LIN PROTOCOL CONTROLLERS

As part of the on-chip peripherals, the available number of integrated UART/LIN protocol controllers typically depends on the performance class of the MCU family. In other words, the higher the performance class, the more LIN channels are integrated. The choice of a high-end/high pin count MCU (bigger footprint) might be necessary even if the higher MCU performance is not needed from an application perspective.

Furthermore, the higher the required number of integrated UART/LIN protocol controllers, the smaller the range of available MCU families. In the worst case, no MCU variant with sufficient UART/LIN protocol controllers is available within the preferred MCU family. Potentially, even a changeover to a new MCU architecture might be unavoidable. This basically means that the number of LIN channels, instead of the required performance, drives the MCU choice.

FLEXIBILITY AND SCALABILITY THROUGH SPI INTERFACE

What if the LIN protocol controller could move into the LIN transceiver? By integrating a LIN protocol controller into a LIN transceiver device, the TXD and RXD signals between MCU and LIN transceiver are eliminated. Instead, a standard MCU interface for peripherals can be used, such as the Serial Peripheral Interface (SPI). Such a LIN device, with on one side an SPI and the other side a LIN bus interface, is called an "SPI-to-LIN bridge."

MCU SELECTION DECOUPLED FROM LIN CHANNEL COUNT

Moving the LIN protocol controller from the MCU into the LIN transceiver decouples the MCU selection from the number of LIN channels. The number of UART/LIN protocol controllers integrated in the MCU is no longer a relevant/limiting factor. In fact, no integrated UART/LIN controller is needed at all; to accomplish LIN protocol communication, the MCU just needs to feature a standard SPI function.

As a SPI-to-LIN bridge decouples the MCU selection from the LIN channel count, the required MCU performance becomes the primary selection criterion rather than the number of LIN communication channels that needs to be handled. This significantly widens the range of suitable and available MCUs.

With SPI, the required number of MCU interface pins becomes almost independent from the number of LIN channels. This pays off the most for higher numbers of LIN channels. For instance, an SPI-to-LIN bridge with four integrated LIN channels would require one additional MCU chip select pin for scaling up from four to eight LIN commander channels. Figure 2 illustrates a MCU pin count example with two SPI-to-Quad LIN bridges.

Depending on the computational footprint of the application, this opens up the use of smaller, mainstream low-pin-count MCUs. Furthermore, an SPI-to-LIN bridge enables upgrading legacy applications with additional LIN channels even though the applied MCU does not provide the extra required UART/LIN protocol controllers.

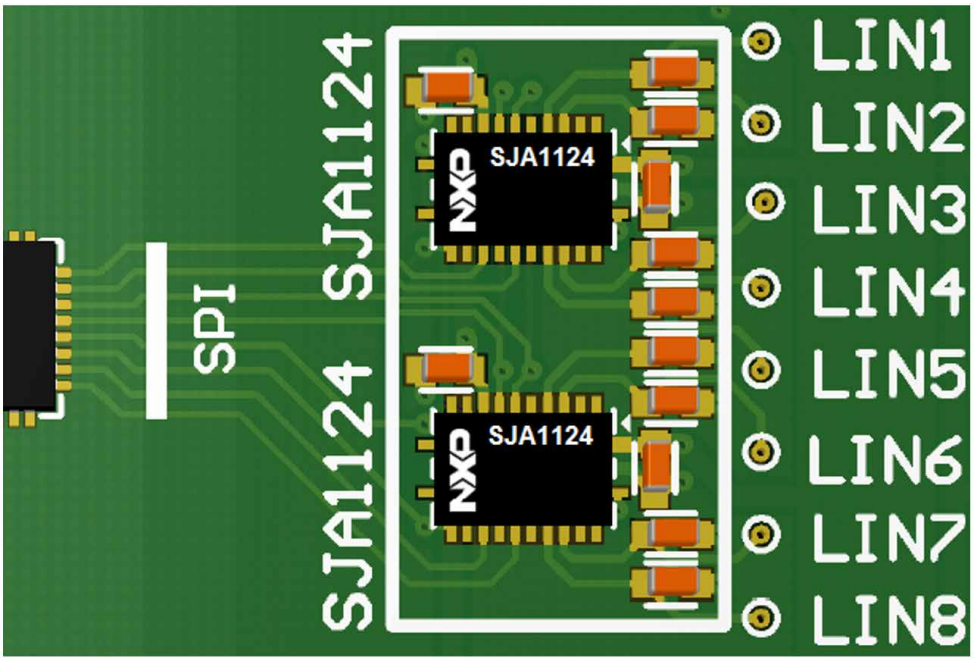

Figure 2: MCU pin count example: eight MCU pins for eight LIN commander channels based on SPI-to-Quad LIN bridge

HARDWARE INDEPENDENT LIN SOFTWARE DRIVER

With the LIN protocol controller moving out of the MCU, the AUTOSAR LIN driver becomes independent from the MCU hardware. Like the AUTOSAR LIN transceiver driver, the LIN driver depends now on the SPI-to-LIN bridge implementation only, using the standard SPI hardware of an MCU. For this reason, the LIN driver of an SPI-to-LIN bridge becomes part of the AUTOSAR Communication Hardware Abstraction.

As illustrated in Figure 3, both LIN Transceiver driver and LIN driver, are located between the AUTOSAR LIN Interface and the SPI handler/driver. As such, it can be either part of the AUTOSAR ECU Abstraction Layer software from a software vendor or an add-on to AUTOSAR Microcontroller Abstraction Layer (MCAL) packages from a microcontroller vendor. This enables fast reuse of the LIN software driver for different MCU implementations.

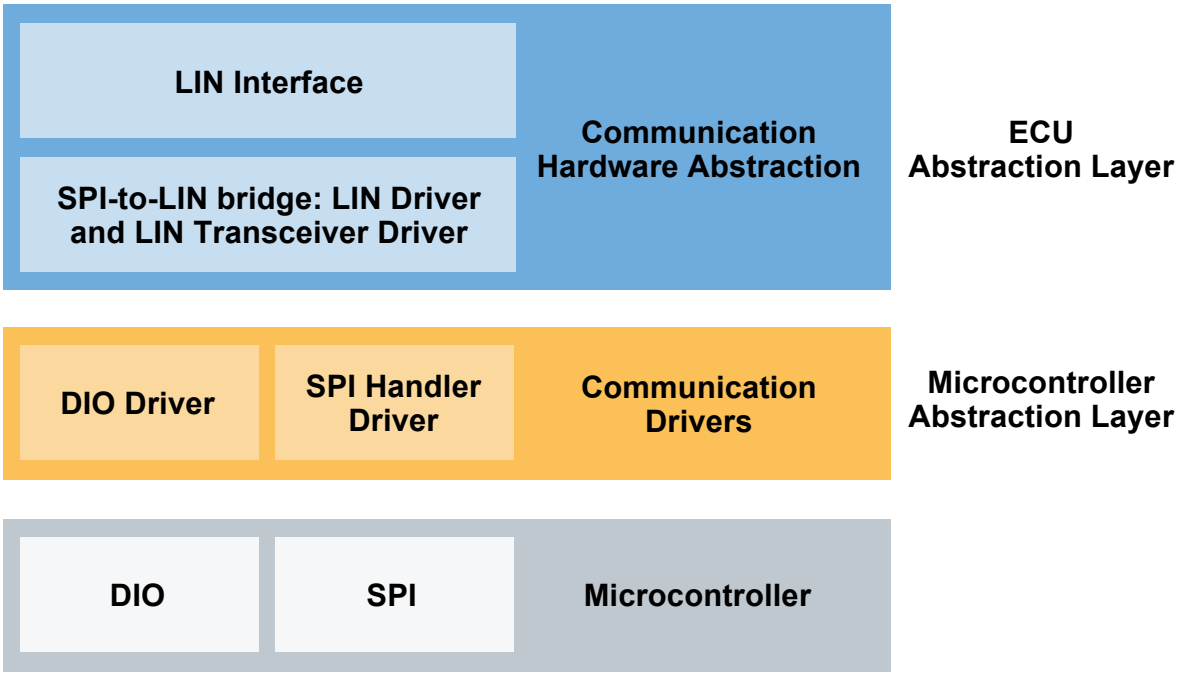

Figure 3: SPI-to-LIN bridge AUTOSAR LIN driver as part of Communication Hardware Abstraction

SPI PERFORMANCE IS KEY

The use of an SPI as MCU interface for multiple LIN channels consequently sets the focus on the SPI performance, because it becomes the bottleneck for the LIN data flow; the SPI runtime is the limiting factor for the data throughput. Therefore, it also limits the maximum number of LIN channels, which can be adequately served by one SPI.

The SPI runtime depends on the chosen hardware (MCU and SPI-to-LIN bridge) and software implementation (SPI handler/driver). Due to the diversity of MCU SPI implementations (optimized for die size, low power or speed, etc.) and SPI software driver implementations (optimized for code size or speed, etc.) a generic maximum number of operable LIN channels per SPI is challenging to provide. Typically, one SPI should be able to serve at least eight LIN channels.

BOARD SPACE INCREASINGLY UNDER PRESSURE

A conventional LIN commander termination is built up with discrete parts, such as one or two pull-up resistors and one diode. This means that a maximum of three discrete parts per LIN commander channel are needed, as highlighted in Figure 4.

In applications with several LIN commander channels (e.g., four LIN channels), the board space consumed by the LIN commander termination may become a constraining factor, especially when the total available board area is already tight. The red frame in Figure 5 shows the LIN commander termination board space for four LIN commander channels.

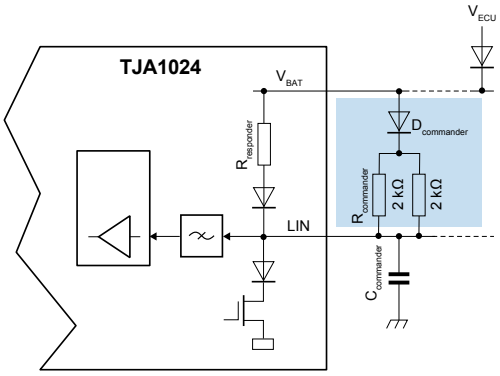

Figure 4: Example application with conventional LIN commander termination

Applications with multiple LIN channels are typically LIN commander applications. For this reason, the integration of the LIN commander termination into the LIN transceiver is a logical step for a multiple LIN channel transceiver.

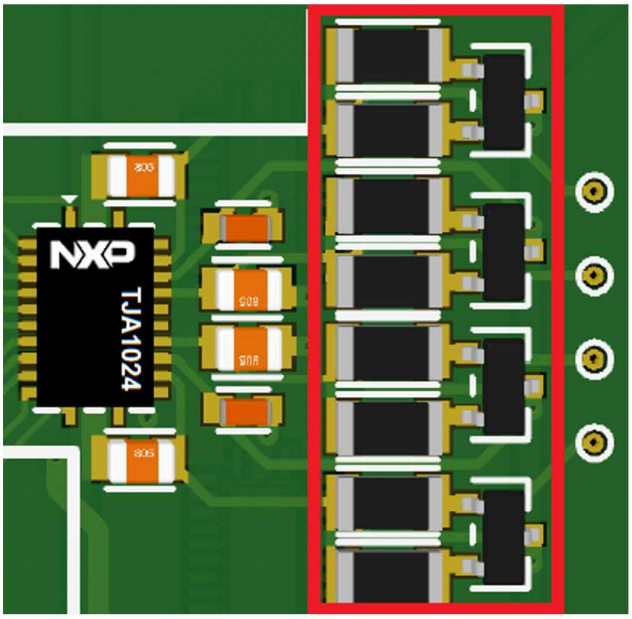

Figure 5: LIN commander termination board space example with four LIN commander channels

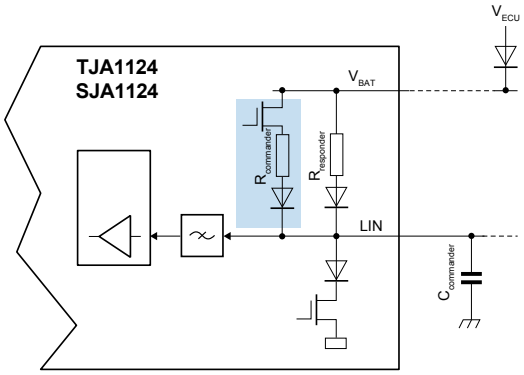

Figure 6: Example with integrated LIN commander termination

A high-side switch is added to the integrated LIN commander termination circuit to cut off the short-circuit current through the LIN commander pull-up resistor for fault cases such as a LIN short-circuit to ground. Figure 6 shows an example in which the integrated LIN commander termination is highlighted.

With an integrated the LIN commander termination, the LIN related board space for applications with several LIN commander channels (e.g., four LIN channels) depends mainly on the number of LIN channels per LIN transceiver and the device package outline. A leadless package with exposed pad combines small size with good thermal connection to the board. Figure 7 shows the board space consumed by a quad LIN transceiver with integrated LIN commander termination.

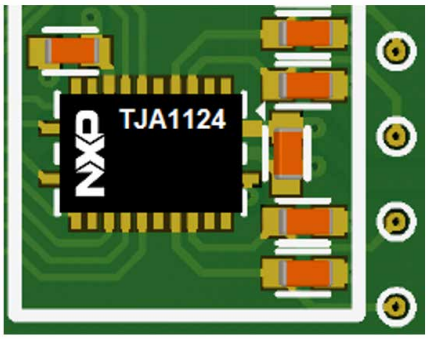

Figure 7 Board space example with 4 LIN commander channels with integrated LIN commander termination

APPLICATION EXAMPLE: SPI-TO-QUAD-LIN BRIDGE BASED ON NXP'S SJA1124

An SPI-to-LIN bridge device is available from NXP. The SJA1124 is an SPI-to-Quad-LIN bridge device incorporating four LIN channels. Each LIN commander channel contains a LIN protocol controller and LIN transceiver with commander termination. LIN data communication is accomplished via SPI; the SJA1124 converts the transmit data stream received on the SPI input into LIN commander frames on the LIN bus. The data stream received on the LIN bus can be read via SPI. A complete LIN frame can be transmitted in one single SPI operation.

As shown in the example in Figure 8, eight LIN channels can be built up with two SJA1124 devices. In this configuration, five MCU pins are needed for the SPI communication (clock, data in, data out, chip select 1 and chip select 2). The on-chip baud rate generator in the LIN protocol controllers requires a reference clock signal. For detection of interrupt events, one shared external interrupt MCU input or two dedicated external interrupt MCU inputs can be optionally connected to the interrupt output of the SJA1124, respectively.

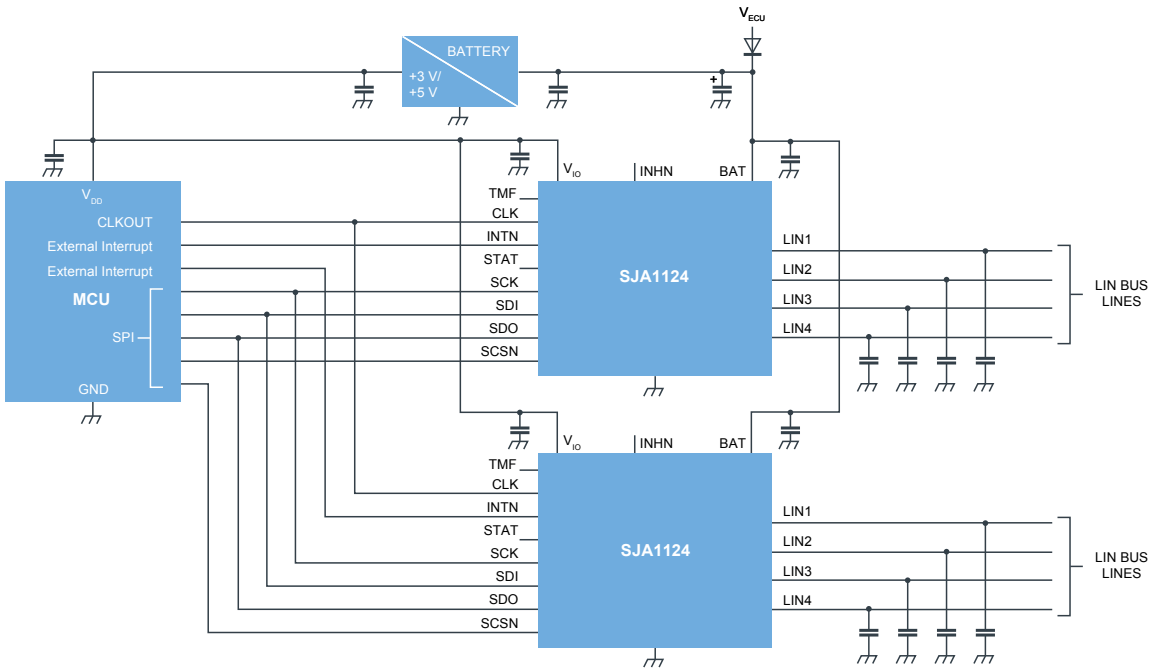

Figure 8: SPI-to-LIN bridge application example with eight LIN channels built up of two SJA1124 from NXP

SUMMARY/CONCLUSION

With the introduction of an SPI-to-LIN bridge with integrated LIN termination, the BOM, board space and costs for multi-channel LIN applications are significantly reduced:

- Fewer discrete components
- Fewer MCU interface signals

Decoupling the MCU from the LIN interface by moving the LIN protocol controller out of the MCU offers unprecedented flexibility:

- Provides the ability to tailor the MCU choice to primary application criteria
- Enhances scalability through MCU independency from LIN channel count
- Enables fast reuse of the SPI-to-LIN bridge software driver for different MCU implementations
- Simplifies redesign of legacy designs if additional LIN connectivity is required

Thus, an SPI-to-LIN bridge like NXP's SJA1124 is a great companion device for MCUs to handle a high number of LIN commander channels, offering unprecedented cost and board space savings in multi-channel LIN applications.

HOW TO REACH US

Home Page: www.nxp.com
Web Support: www.nxp.com/support

USA/Europe or Locations Not Listed:
NXP Semiconductors USA, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.nxp.com/support

Europe, Middle East, and Africa:
NXP Semiconductors Germany GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.nxp.com/support

Japan:
NXP Japan Ltd.
Yebisu Garden Place Tower 24F,
4-20-3, Ebisu, Shibuya-ku,
Tokyo 150-6024, Japan
0120 950 032 (Domestic Toll Free)
www.nxp.com/jp/support/

Asia/Pacific:
NXP Semiconductors Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@nxp.com

www.nxp.com

Document Number: SJA1124WP REV 1