

AN10452

Interfacing Philips Bridge IC with Philips microcontroller

Rev. 01 — 17 May 2006

Application note



Document information

Info	Content
Keywords	SC16IS750 to P89LPC935, Bridge IC to microcontroller, serial interface
Abstract	The operation and description of Philips LPC900 Series microcontroller such as P89LPC935 to Philips Bridge IC such as SC16IS750 for high-speed serial data communication is discussed in this application note. In addition, the source code in C language, containing communication routines between the P89LPC935 microcontroller and the SC16IS750 Bridge IC is provided. This application note is also applicable to other Bridge ICs such as SC16IS740, SC16IS760, SC16IS752, and SC16IS762.

PHILIPS

Revision history

Rev	Date	Description
01	20060517	application note; initial version

Contact information

For additional information, please visit: <http://www.semiconductors.philips.com>

For sales office addresses, please send an email to: sales.addresses@www.semiconductors.philips.com

1. Introduction

The Philips Bridge IC is a new generation of interface solutions for managing high-speed serial data communication among various bus interfaces such as SPI, I²C-bus, and UART including RS-232 and RS-485. The Bridge IC is commonly used to overcome the limitation of the host bus interface to peripherals and to provide easy interface with existing different serial buses.

The description of the block diagram, hardware, firmware, and software are described in the next paragraphs for users to quickly understand the implementation of the Philips LPC900 Series microcontroller to Philips Bridge IC serial interface for RS-232 point-point communication, RS-485 multi-drop application, IrDA wireless links communication, and GPIO interface. The source code in C language is provided to show how to write a simple communication program between the microcontroller and the Bridge IC serial interface. The goal is to help users to design the Bridge IC in their application and also shorten their product development cycle.

2. Block diagram

The block diagram depicted in [Figure 1](#) shows the circuit connection between a Philips Bridge IC such as SC16IS750 and Philips LPC900 Series microcontroller such as P89LPC935. The Bridge IC offers simple, flexible, and minimal connection to the microcontroller. The Bridge IC embedded SPI and I²C-bus for the host interface so the microcontroller can easily control the Bridge IC with four-wire connection through SPI or I²C-bus, which is a widely used serial bus interface. If the SPI-bus interface is selected, the Bridge IC can interface to the microcontroller with a 4-wire connection. The signals on the 4 wires are MISO (Master Input Slave Output), MOSI (Master Output Slave Input), CS (Chip Select), and SPICLK (SPI Clock). If the I²C-bus is selected, the Bridge can interface to the microcontroller with a 2-wire connection. The signals on the 2 wires are SCL (Serial Clock) and SDA (Serial Data).

After the microcontroller and Bridge IC are wired properly and their connection is established, the microcontroller can send data to and receive data from UART devices via the Bridge IC. The Bridge IC receives the data from the microcontroller via the SPI or I²C-bus interface and sends the data to the UART devices via RS-232 or RS-485 bus interface, and when the Bridge IC receives the data from the UART devices the Bridge IC will notify the microcontroller by generating an interrupt signal; then the microcontroller can access the data from the Bridge IC via the SPI or I²C-bus interface.

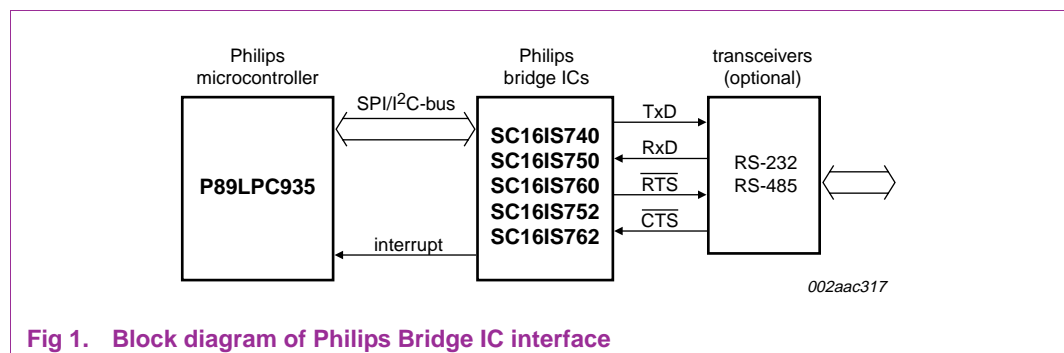


Fig 1. Block diagram of Philips Bridge IC interface

3. Hardware description

The hardware consists of the two major parts, which are the microcontroller and the Bridge IC. The Bridge IC provides a seamless interface convergence between the SPI or I²C-bus and RS-232 or RS-485 protocols and vice versa. In addition, the Bridge IC checks the data transmission for errors and maintains very high data throughput.

The functions of the two major parts are described as follows:

- Philips P89LPC935 microcontroller

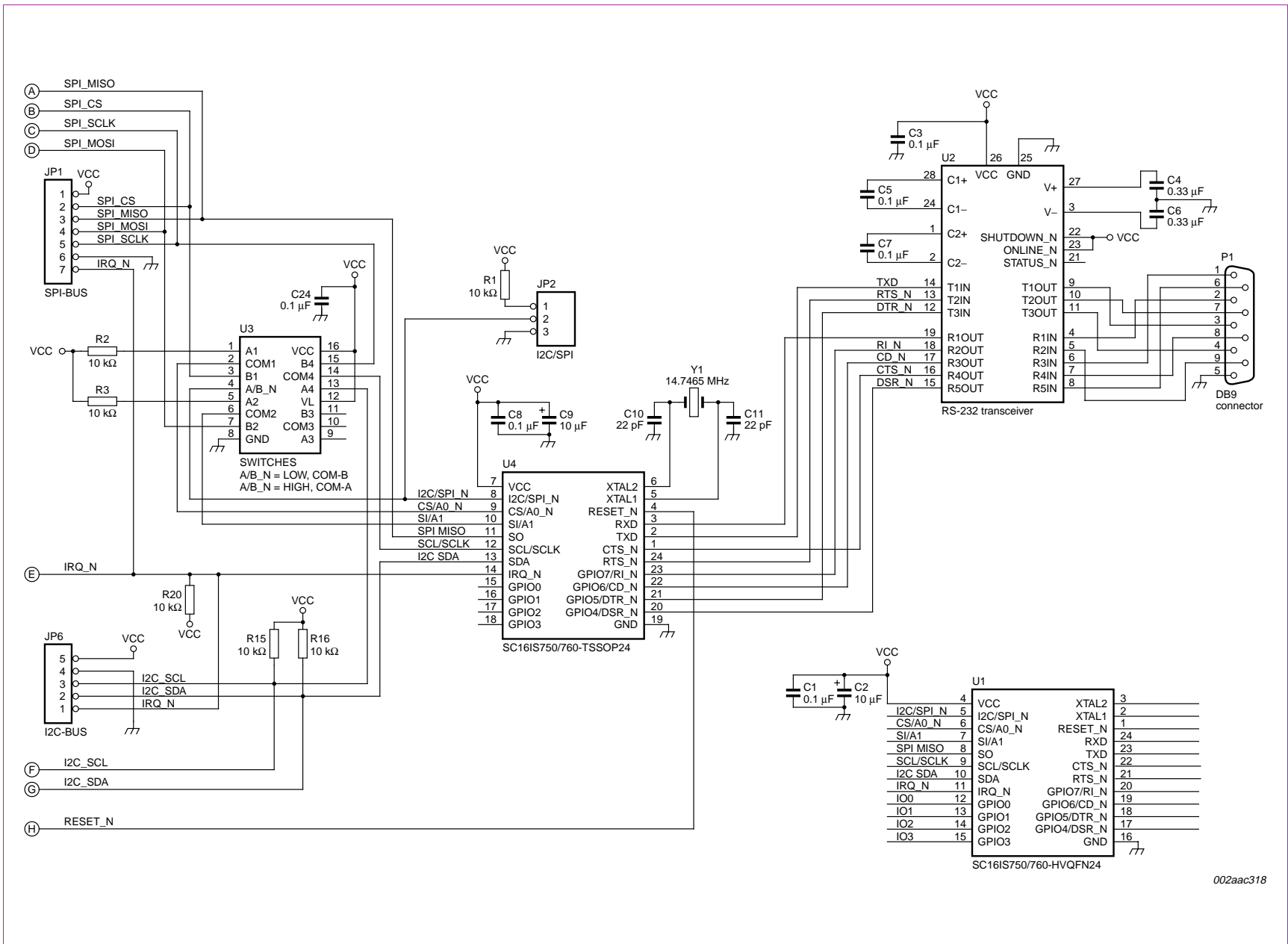
The microcontroller connection to the Bridge IC is through the SPI or I²C-bus interface. The microcontroller acts as a master controlling the Bridge IC with the embedded firmware code in the microcontroller ROM (Read Only Memory). The microcontroller initiates the data transfer on the bus and controls the Bridge IC with the chip select pin for SPI-bus interface.

- Philips SC16IS750 Bridge IC

The Bridge IC acts as a slave interface convergence between the microcontroller and UART peripheral. The Bridge IC handles the SPI or I²C-bus interface to the microcontroller and the UART that consists of transmitter and receiver. The transmitter sends the data received from the microcontroller to the UART peripherals. The receiver sends the data received from the UART peripheral to the microcontroller.

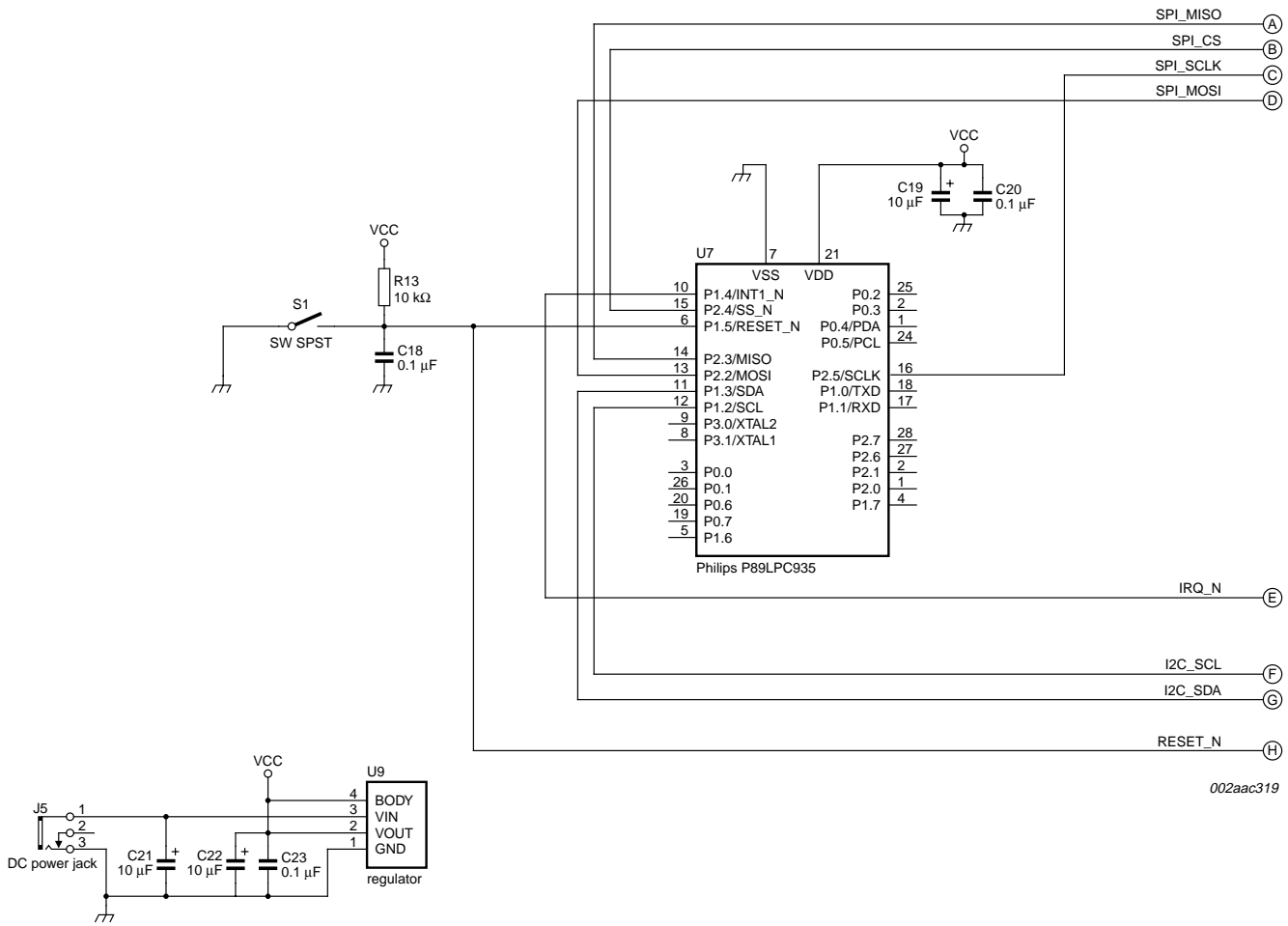
The transceivers are optional and can be intended for RS-232 or RS-485 communication interface. The transceivers consist of drivers and receivers. The drivers convert the UART-logic output levels to RS-232 or RS-485 signals, whereas the receivers convert the RS-232 or RS-485 signals to UART-logic input levels. If the UART-logic level conversion is not required, the transceivers are optional.

For hardware connection details, please refer to the schematic shown in [Figure 2](#) and [Figure 3](#).



002aac318

Fig 2. Schematic diagram (part 1)



002aac319

Fig 3. Schematic diagram (part 2)

4. Software description

The programming of the Bridge IC can be done by writing firmware code, which requires the following software:

- Keil is one of the embedded system vendors that provide the software development tools for Philips P89LPC935 microcontroller. The software compiles the firmware code written in C and generates an 'Intel Hex' file. The software evaluation development kit can be downloaded from the Keil website.
- "Flash Magic" is free Windows application software that allows easy programming of Philips P89LPC935 microcontroller. The software loads the 'Intel Hex' file to the microcontroller by using its in-system programming mode communicating through serial port. The software can be downloaded from the Embedded Systems Academy website under "Flash Magic".

5. In-System Programming mode

Philips P89LPC935 microcontroller has an on-chip Flash program memory with ISP (In-System Programming) that allows the microcontroller to be programmed without removing the microcontroller from the board, and also the microcontroller, which if previously programmed can be reprogrammed without removal from the board.

The microcontroller must be powered up in a special 'ISP mode' to perform the ISP operation. The ISP mode allows the microcontroller to communicate with a host device such as PC through a serial port. The host sends commands and data to the microcontroller. The commands can be erase, read, and write. After the completion of the ISP operation, the microcontroller is reconfigured and has to be reset or power cycled so the microcontroller will operate normally.

The ISP programming for the device can be done using a Windows application software, which uses an Intel Hex file as input to program it. For more information about the software, please refer to [Section 4 "Software description"](#).

6. Firmware description

The firmware code for the P89LPC935 microcontroller is written in C language. It can be compiled by using an embedded C compiler. For more information about the compiler, please refer to [Section 4 "Software description"](#).

The firmware code consists of three major blocks: Main Loop, Interrupt Service Routine, and Bus Interface layer, and are described in

6.1 Main Loop

The function of the Main Loop is to reset and initialize the Bridge IC by writing a character to the Bridge IC' register. Inside the Main Loop, the microcontroller can select one of the two methods for communicating to the Bridge IC. The first method is polling the Bridge IC' status register regularly. The second method is using an interrupt handler in the Interrupt Service Routine until the Bridge IC generates an interrupt. If using the interrupt handler,

the microcontroller and the bus interrupt bits must be enabled. The other function of the Main Loop is to keep checking the event flags and pass to the appropriate subroutine for further processing.

6.2 Interrupt Service Routine

The microcontroller uses the Interrupt Service Routine (ISR) to handle an interrupt generated by the Bridge IC. As soon as the Bridge IC generates an interrupt, the ISR checks the interrupt status of the Bridge IC to determine the interrupt type and sets up proper event flags to inform the Main Loop program for processing the interrupt request.

6.3 Bus interface layer

The bus interface layer handles the SPI or I²C-bus interface between the microcontroller and the Bridge IC. The three functions in the bus interface layer are:

- **BusInterface_Init:** The microcontroller initialize the SPI or I²C-bus controller to initiate the SPI or I²C-bus data transfer.
- **BusInterface_Read:** The microcontroller reads data from the SPI or I²C-bus and stores the data for further processing.
- **BusInterface_Write:** The microcontroller writes data to the SPI or I²C-bus for the Bridge IC to read.

7. Conclusion

Philips Bridge IC provides easy interface to a host controller such as Philips P89LPC935 microcontroller, enables seamless and high-speed SPI or I²C-bus to RS-232 or RS-485 protocols convergence including GPIO for general-purpose input/output and IrDA for wireless links, and offers low voltage operation, low power consumption, and compact design which is suitable for battery-operated applications. In addition, the Bridge IC reduces software overhead, frees up the host controller resources, increases design flexibility, and improves overall system performance. For more details about the Bridge ICs, please download the data sheets from the Philips Semiconductors website under "Interface & Control, UARTs".

8. Legal information

8.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. Philips Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

8.2 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, Philips Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — Philips Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — Philips Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a Philips Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Philips Semiconductors accepts no liability for inclusion and/or use of Philips Semiconductors products in such equipment or applications and therefore such inclusion and/or use is for the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

8.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of Koninklijke Philips Electronics N.V.

9. Contents

1	Introduction	3
2	Block diagram	3
3	Hardware description	4
4	Software description	7
5	In-System Programming mode	7
6	Firmware description	7
6.1	Main Loop	7
6.2	Interrupt Service Routine	8
6.3	Bus interface layer	8
7	Conclusion	8
8	Legal information	9
8.1	Definitions	9
8.2	Disclaimers	9
8.3	Trademarks	9
9	Contents	10



Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© Koninklijke Philips Electronics N.V. 2006. All rights reserved.

For more information, please visit: <http://www.semiconductors.philips.com>.
For sales office addresses, email to: sales.addresses@www.semiconductors.philips.com.

Date of release: 17 May 2006

Document identifier: AN10452_1