Semiconductors

AN10339

UART serial interface through USB evaluation board

Rev. 01 — 18 February 2005 Application note

Document information

Info	Content
Keywords	USB to UART, UART evaluation board, USB serial interface
Abstract	The operation and description of the UART serial interface through USB is discussed in this application note. In addition, the firmware application code in C language, containing communication routines of the UART serial interface through USB is provided.

PHILIPS

Revision history

Rev	Date	Description
01	20050218	Application note (9397 750 14361).

© Koninklijke Philips Electronics N.V. 2005. All rights reserved.

1. Introduction

The UART serial interface through USB evaluation kit is a useful development tool. It comes with an evaluation board, firmware application code, and complete documents. The kit enables users to quickly understand the implementation of UART serial interface through USB, and UART to RS-232 serial port interface.

The description of the block diagram, hardware, software, and firmware are described in the next paragraph. The firmware application code in C language is provided to show how to write a simple communication program of UART serial interface through USB. The goal is to help users to shorten their product development cycle and to evaluate the Philips Dual UART functionality.

2. Block diagram

The block diagram depicted in Figure 1 shows the bus interface of UART serial interface through USB. The USB has a general-purpose parallel interface for communication with various microcontrollers such as the 8051 microcontroller. The 8051 microcontroller sends data to, and receives data from, the UART through the 8-bit data bus. Then, the UART converts the parallel data to serial data and transmits the serial data to the serial port through the RS-232 line drivers, whose function is to convert the logic voltage levels. Also, the UART receives the serial data from the serial port through the RS-232 line receivers and converts the serial data to an 8-bit data. Then the UART sends the parallel data back to the microcontroller, and the microcontroller transmits the data to the USB interface. As soon as the host PC requests for the data, the USB interface will pass the data through USB bus protocol communication.

Any wireless modules such as Bluetooth and GPS can be attached to the UART serial interface for having serial communication between the modules and a host PC through USB, and also between one module and another module(s). For details about the interface between Philips Dual UART and a Bluetooth module, please refer to application note *AN10307*. Also, the application note *AN10219* shows how to use Philips UARTs to implement an IrDA interface. The application notes can be found at the Philips Datacom website: http://www.standardproducts.philips.com/support/appnotes/datacom/.

© Koninklijke Philips Electronics N.V. 2005. All rights reserved.

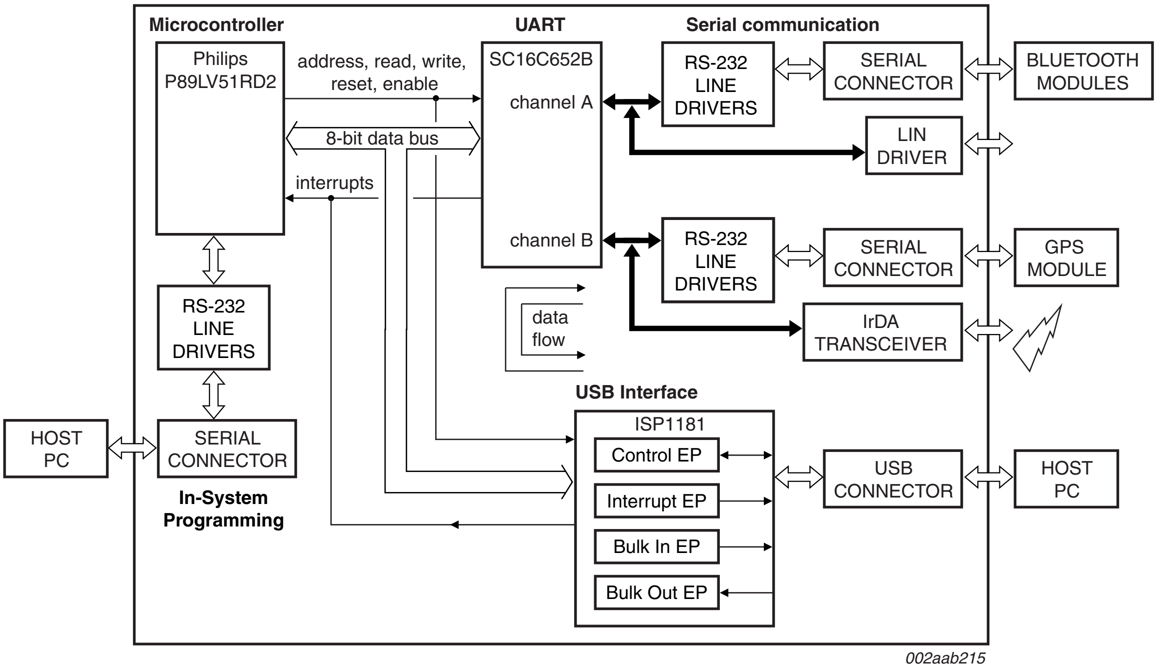

Fig 1. Block diagram of UART serial interface through USB evaluation board

© Koninklijke Philips Electronics N.V. 2005. All rights reserved.

3. Hardware description

The evaluation board consists of four major parts described in the next paragraph. The board behaves as a UART serial interface through USB bridge controller. It provides a function to receive and transmit data between modules and a host PC, and also between one module and another module(s). In addition, it provides an in-system programming function to download firmware application code to the microcontroller. After power-on reset, the operation of the bridge controller will be based on the programming of the firmware application code.

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

- Philips P89LV51RD2 is an 8051 microcontroller. The microcontroller connection to UART and USB controllers is through the 8-bit data bus and control signals—address, write, read, reset, and chip select signals. The microcontroller acts as a host controlling the USB and UART devices on the board with the embedded firmware application code. The firmware application code controls the job of the microcontroller, which are:
  - controls the operation of all devices on the board
  - handles the data streaming between USB and UART
  - configures USB and UART registers
  - handles USB enumeration.
- Philips SC16CxxxB is a dual-channel UART device with 32-byte FIFOs and IrDA encoder/decoder. It interfaces between the microcontroller and RS-232 line drivers. The UART device mainly consists of two sections: transmitter and receiver. The transmitter performs parallel-to-serial conversion on the 8-bit data transmitted from the microcontroller. The receiver performs serial-to-parallel conversion on the serial data received from a peripheral device.
- Philips RS-232 drivers/receivers are the transceivers. The first one allows the microcontroller to be in-system programmed through a PC’s serial port such as COM1. The second one allows the UART to be in serial communication with a peripheral device through another PC’s serial port such as COM2. The RS-232 mainly consists of two sections: drivers (transmitters) and receivers. The drivers convert the CMOS-logic output levels to RS-232 signals, whereas the receivers convert the RS-232 signals to CMOS-logic output levels.
- Philips ISP1181B is a full-speed USB interface device with a high-speed general-purpose parallel interface for communication with many kinds of microcontroller. In the evaluation board, the ISP1181B appears as an I/O device with 8-bit data bus and 1-bit chip selection line. It has endpoints used to transfer data or command directly to/from the UART device.

© Koninklijke Philips Electronics N.V. 2005. All rights reserved.

4. Software description

The UART serial interface through USB evaluation board is software-based programming. The programming of the board can be done by writing firmware application code, which requires the following software:

- Raisonance is one of the embedded system vendors that provide the development tools for the 8051 microcontroller. The software compiles the firmware application code in C and generates an 'Intel® Hex' file. The Raisonance free evaluation development kit can be downloaded from http://www.raisonance.com.
- Flash Magic is a free Windows application software that allows easy programming of Philips Flash Microcontrollers. The software loads the 'Intel Hex' file to the microcontroller by using its in-system programming mode communicating through a serial port such as COM1. The software can be downloaded from http://www.esacademy.com/software/flashmagic.

5. In-System Programming (ISP) mode

Philips P89LV51RD2 microcontroller has an on-chip Flash program memory with ISP, which allows the microcontroller to be programmed without removing the microcontroller from the board. Also, the microcontroller 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 a PC through a serial port. The host sends commands and data to the microcontroller. The commands can be: erase, read, verify, 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 execute the updated firmware application code and operate normally.

The ISP programming fro the device can be done using a Windows application software which uses an Intel Hex file as input to program it. The Intel Hex file is generated after compiling the firmware application code. For more information about the software, please refer to Section 4 "Software description".

6. Firmware description

The firmware application code for the evaluation board is written in C language. It can be compiled by using an embedded C compiler for 8051-based microcontroller. For more information about the software compiler, please refer to Section 4 "Software description".

The firmware application code consists of four major blocks, which are: Main Loop, Interrupt Service Routine, USB Protocol layer, and UART layer.

6.1 Main Loop

The function of the Main Loop is to reset and initialize the 16C UART and the ISP1181 USB devices by writing a character (an 8-bit data) to the UART and the ISP1181 registers. Inside the Main Loop, the microcontroller can select one of the two methods programmed for communicating to the UART. The two methods are: polling the register status of the

© Koninklijke Philips Electronics N.V. 2005. All rights reserved.

UART regularly, and using an interrupt handler in the interrupt service routine. If using the interrupt handler, the microcontroller and the UART interrupt bits must be enabled. The other function is to keep checking the event flags and pass to the appropriate subroutine for further processing.

6.2 Interrupt Service Routine (ISR)

The microcontroller uses the interrupt service routine to handle an interrupt generated by the 16C UART and the ISP1181 USB devices. As soon as the UART or the ISP1181 generates an interrupt, the ISR checks the interrupt status of the UART or the ISP1181 to determine the interrupt type and sets up proper event flags to inform the Main Loop program for processing the interrupt.

6.3 USB protocol layer

The USB protocol layer handles standard USB device requests such as ‘get descriptor’, ‘set configuration’, ‘set interface’, and ‘set address’ which is defined in Chapter 9 of *USB specification 1.1*. The requests are used for configuring a USB peripheral device, controlling the stat of its interface along with other miscellaneous features. The host issues device requests using control transfer mechanism. For details, please refer to Chapter 9 ‘Protocols’ in the *USB specification 1.1*.

In addition, the USB protocol layer handles specific vendor requests, such as ‘set baud rate’, ‘set data bit’, and ‘set stop bit’, which is defined specifically for the 16C UART device. The request are used for configuring the UART device by initializing the UART registers.

6.4 UART layer

The UART layer handles the bus interface between the microcontroller and the UART. The three functions in the UART layer are:

- UART_Reset. The microcontroller resets the UART device.
- UART_Read. The UART places a character on the data bus for the microcontroller to read and the microcontroller stores the character.
- UART_Write. The microcontroller places a character on the data bus for the UART to read.

© Koninklijke Philips Electronics N.V. 2005. All rights reserved.

7. Conclusion

The use of the evaluation kit is for the users who want to understand how Philips Dual UART is interfacing with an 8-bit microcontroller such as 8051 and the implementation of Philips UART serial interface through USB. The evaluation board provides a function to control the data transfer of UART serial interface through USB. The board shows the circuit connection of the UART to the microcontroller, the UART to RS-232 transceivers, and the UART serial interface through USB. The provided firmware application code in C language shows the serial communication protocol of the UART serial interface through USB. Also, it shows how to access UART and USB devices' control and data registers.

The evaluation board allows the connection of any wireless modules such as Bluetooth and GPS which has a UART serial interface, for having serial communication between the modules and a host PC through USB, and also between one module and another module. With a growing trend to faster data rates, Philips' UART solutions provide the speed and simplicity to interface with today's and tomorrow's wireless solution.

More information on Philips' UART solutions (SC16CxxxB and SC28Lxx family of Advanced UARTs) can be found at http://www.standardlogic.philips.com/datacom.

© Koninklijke Philips Electronics N.V. 2005. All rights reserved.

8. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status ‘Production’), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

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

9. Trademarks

Intel — is a registered trademark of Intel Corporation.
Bluetooth — The Bluetooth® word mark and logos are owned by the Bluetooth SIG, Inc., and any use of such marks by Philips Semiconductors is under license. Other trademarks and trade names are those of their respective owners.

© Koninklijke Philips Electronics N.V. 2005. All rights reserved.

10. Contents

© Koninklijke Philips Electronics N.V. 2005

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Date of release: 18 February 2005
Document number: 9397 750 14361

Published in The Netherlands