



# AN10319

## 8051 microcontroller to UART serial interface evaluation board

Rev. 01 — 26 August 2004

Application note

### Document information

Info	Content
<b>Keywords</b>	8051 microcontroller to UART, UART evaluation board, UART serial interface
<b>Abstract</b>	The operation and description of the 8051 microcontroller to UART serial interface is discussed in this application note. In addition, information for obtaining the source code in C language, containing communication routines between the 8051 core microcontroller and the UART is provided.

## Revision history

Rev	Date	Description
1	20040826	Application note; initial version (9397 750 13533).

## 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](mailto:sales.addresses@www.semiconductors.philips.com)

## 1. Introduction

The UART Serial Evaluation kit is a useful development tool. It comes with a demo board, firmware source code, and complete documents. The kit enables users quickly to understand the implementation of a microcontroller such as 8051 to UART serial interface and the UART to RS-232 serial port interface.

The description of the block diagram, hardware, software, and firmware are described in [Section 2](#). The source code in C language is provided to show how to write a communication program between the 8051 microcontroller and the UART. The goal is to help users to shorten their product development cycle.

## 2. Block diagram

The block diagram depicted in [Figure 1](#) shows the bus interface between a UART and a microcontroller such as 8051. The 8051 microcontroller sends data to and receives data from the UART through the 8-bit parallel 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 which function is to convert the CMOS logic to TTL logic. 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 parallel data. Then, the UART sends the parallel data to the microcontroller.

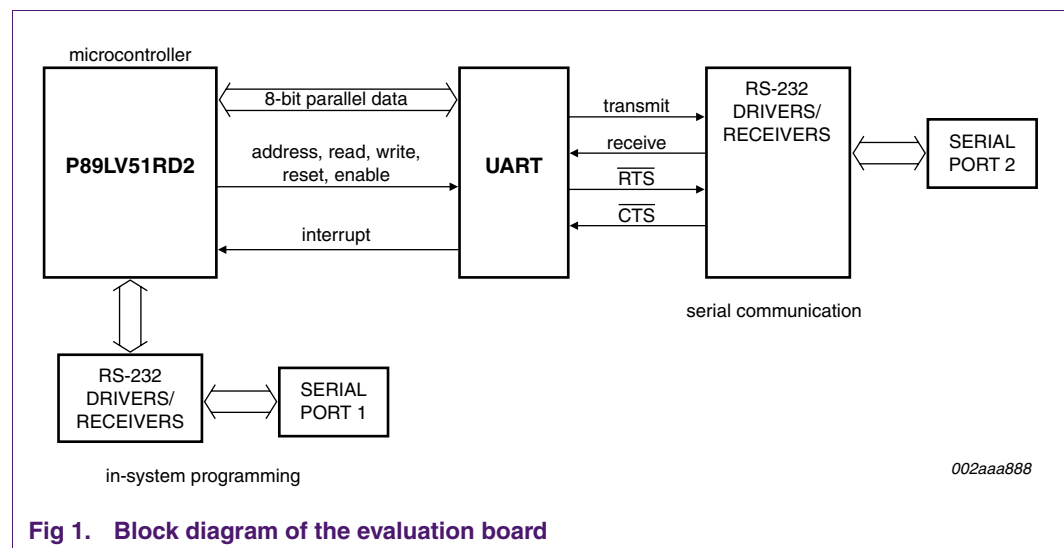


Fig 1. Block diagram of the evaluation board

### 3. Hardware description

---

The evaluation board consists of three major parts and provides a function to convert the parallel data bus to RS232 serial port and vice versa. Also, it provides an in-system programming function.

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

- Philips P89LV51RD2 is an 8051 microcontroller. The microcontroller connection to the UART is through the 8-bit parallel data bus and control signals - address, write, read, reset, and chip select signals. The microcontroller acts as a host controlling the UART device on the board with the embedded firmware.
- Philips SC16Cxxx is a UART device interfacing between the microcontroller and RS-232. The UART mainly consists of two sections: transmitter and receiver. The transmitter performs parallel-to-serial conversion on data transmitted from the microcontroller. The receiver performs serial-to-parallel conversion on 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.

### 4. Software description

---

The programming of the demo board can be done by writing firmware code 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 code 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 serial port. The software can be downloaded from <http://www.esacademy.com/software/flashmagic>.

### 5. In-system programming mode

---

Philips P89LV51RD2 microcontroller has an on-chip Flash program memory with ISP (In-System Programming), which allows the microcontroller to be programmed without removing the microcontroller from the board and also the microcontroller, which 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 evaluation board 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 UART layer. Each are described below.

### 6.1 Main Loop

The function of the Main Loop is to reset and initialize the UART by writing a character (an 8-bit data) to the UART register. Inside the Main Loop, the microcontroller can select one of the two methods for communicating to the UART. The two methods are polling the 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 UART. As soon as the UART generates an interrupt, the ISR checks the interrupt status of the UART to determine the interrupt type and sets up proper event flags to inform the Main Loop program for processing the interrupt.

### 6.3 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.

## 7. Conclusion

---

The use of the evaluation kit allows the user to understand how the Philips UART is interfacing with a microcontroller such as 8051. The board shows the circuit connection between the microcontroller to the UART and the UART to RS-232 transceivers. The source codes *Mainloop.c*, *UART.C*, and *UART.H* show the communication protocol between the microcontroller and the UART. They can be accessed at <http://www.standardproducts.philips.com/support/datacom/>.

## 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 license 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.

## 9. Contents

---

<b>1</b>	<b>Introduction</b> .....	<b>3</b>
<b>2</b>	<b>Block diagram</b> .....	<b>3</b>
<b>3</b>	<b>Hardware description</b> .....	<b>4</b>
<b>4</b>	<b>Software description</b> .....	<b>4</b>
<b>5</b>	<b>In-system programming mode</b> .....	<b>4</b>
<b>6</b>	<b>Firmware description</b> .....	<b>5</b>
6.1	Main Loop .....	5
6.2	Interrupt Service Routine (ISR) .....	5
6.3	UART layer .....	5
<b>7</b>	<b>Conclusion</b> .....	<b>6</b>
<b>8</b>	<b>Disclaimers</b> .....	<b>7</b>



© Koninklijke Philips Electronics N.V. 2004

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: 26 August 2004  
Document order number: 9397 750 13533

Published in the U.S.A.