Semiconductors

AN10307

UART to Bluetooth interfacing

Rev. 02 — 11 August 2004 Application note

Document information

Info	Content
Keywords	UART, Bluetooth, wireless
Abstract	This application note shows how a Bluetooth wireless solution can be integrated into a system using a UART from Philips Semiconductors.

PHILIPS

Revision history

Rev	Date	Description
_2	20040811	Application note (9397 750 13929). Section 2.3.2 "Interrupt Enable Register (IER)" re-written.
_1	20040628	Application note, initial version (9397 750 13283).

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

Bluetooth® wireless technology is becoming a popular standard in the communication arena, and it is one of the fastest growing fields in the wireless technologies. It is convenient, easy to use and has the bandwidth to meet most of today's demands for mobile and personal communications.

Devices equipped with Bluetooth technology support wireless point-to-point connections, as well as wireless access to LAN, mobile phone network, the ethernet and home networking. Bluetooth technology handles the wireless part of the communication channel; it transmits and receives data wirelessly between these devices. It delivers the received data and receives the data to be transmitted to and from a host system through a host controller interface (HCI). The most popular host controller interface today is either a UART or a USB link; see Figure 1.

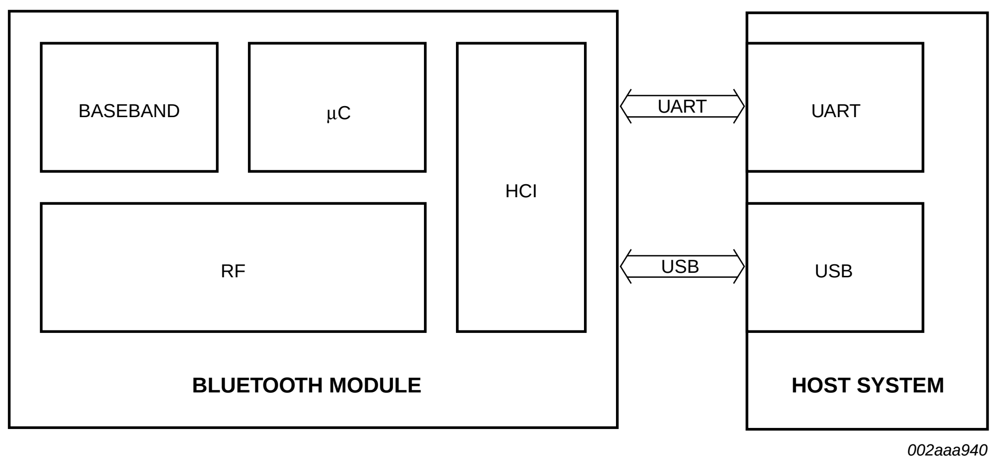

Fig 1. Glueless Bluetooth wireless technology interface

While the host controller interface can be either UART or USB, the simpler transport protocol of the UART results in much less software overhead, more cost-effective hardware solution, and with a high performance UART (such as a UART from Philips Semiconductors) the data throughput on the UART interface is almost comparable to the USB interface.

According to In-Stat/MDR report published in April 2004[1], beyond mobile phones, Bluetooth enabled applications will encompass cordless phones, access points, desktops, notebooks, PC and memory cards, printers, digital camcorders and cameras, a broad variety of consumer equipment (games, speakers, set-top boxes and MP3 players), automotive and industrial.

It is expected that a large percentage of the Bluetooth chips designed into the applications stated above will include a UART in the host-controller. For these configurations, this application note is intended to highlight the benefits that Philips' standalone UARTs can bring into a design.

1. In-Stat/MDR report, April 2004 by Joyce Putscher, *Bluetooth 2004: Poised for Mainstream*.

In this application we will only focus on the UART interface, it can be easily show how a Bluetooth module can be integrated on to a host system through a glueless UART connection and provide the designer an optimal solution for Bluetooth enabled systems.

2. UART

A Universal Asynchronous Receiver and Transmitter (UART) is used for communication with serial input and serial output devices. Serial transmission reduces to cost and complexity of the wirings at the expense of speed, and for many applications this is a desired trade-off.

UARTs provide serial asynchronous receive data synchronization, parallel-to-serial and serial-to-parallel data conversion for both the transmitter and receiver sections. These functions are necessary for converting the serial data stream into parallel data that is required with digital systems. Synchronization for the serial data stream is accomplished by adding start and stop bits to the transmit data to form a data character. Data integrity is insured by attaching a parity bit to the data character. The parity bit is checked by the receiver for any transmission bit errors.

To a host system, the UART appears as an 8-bit input and output port that it can read from and write to. Whenever the host has data to be sent, it just sends these data to the UART in byte format (8-bit wide), whenever the UART receives data from another serial device it will buffer these data in its FIFO (again 8-bit wide), then it will indicate the availability of these data to the host through an internal register bit, or through a hardware interrupt signal.

In addition to the transmitter and the receiver, UARTs from Philips Semiconductors also incorporate other features that significantly reduce software overhead and increase system efficiency. Some of these features are:

- Wide range of supply voltage: 2.5 V, 3.3 V, 5.0 V
- Hardware and software autoflow control
- Large FIFOs (up to 256 bytes)
- Fast baud rate (5 Mbit/s max.)
- Industrial temperature range: –40 °C to +85 °C
- Fast bus access time (43 ns)
- Sleep mode, where the device current consumption is reduced to about 50 μA
- Small footprint (HVQFN32).

Hardware and software autoflow control prevent FIFO overflow conditions automatically. Without automatic flow control, the host software needs to empty the receive FIFO immediately when it is about to be filled up.

Large FIFOs reduce the host processor service time to the UART; this allows the processor more time to do other tasks. Faster baud rate and faster bus access improve the overall system performance; the system can send/receive more data in less time. Figure 2 shows an internal block diagram of a SC16C650B UART.

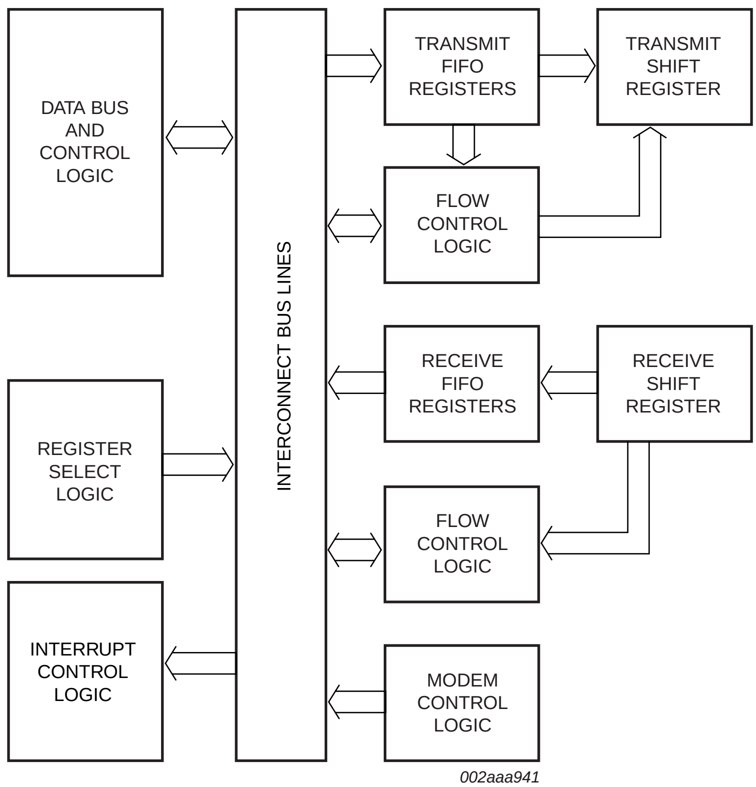

Fig 2. SC16C650B UART internal blocks.

2.1 Data bus and control logic block

The host controller sends and receives data to and from the UART through this block. The control logic block within this block generates various control signals for the internal interconnect bus.

2.2 Register select logic

The Register Select Logic block decodes the addresses from the host to select which UART's internal register the host wants to access.

2.3 Internal registers

The host and the UART communicate through a set of registers. These registers function as data holding registers (THR/RHR), interrupt status and control registers (IER/ISR), a FIFO control register (FCR), line status and control registers (LCR/LSR), modem status and control registers (MCR/MSR), programmable data rate (clock) control registers (DLL/DLH), and a user-accessible scratch pad register (SPR).

2.3.1 Transmit Holding Register (THR) and Receive Holding Register (RHR)

These registers are used to store transmitting and receiving data. The host writes data to THR for transmission, and reads RHR for data received by the UART.

2.3.2 Interrupt Enable Register (IER)

Interrupt Enable Register is used to enable/disable different types of interrupts supported by the UART. Some of these interrupts are Receive Data Ready, Transmit Empty, Line Status Register, and Modem Status Register.

2.3.3 FIFO Control Register (FCR)

FCR is used for enabling the FIFOs, clearing the FIFOs, setting transmitter and receiver trigger levels.

2.3.4 Line Control Register (LCR)

LCR is used to set the data communication format. The word length, number of stop bits, and parity type are selected by writing the appropriate bits to the LCR. Line Status Register (LSR) provides the status of the data transfer between the UART and a remote UART. This register reports framing error, parity error, overrun error and other FIFOs status.

2.3.5 Modem Control Register (MCR)

MCR controls the interface with the modem, data set, or peripheral devices that is emulating the modem. Modem Status Register (MSR) provides information about the current state of the control lines from the modem, data set, or peripheral device to the processor.

2.3.6 Divisor Latch Low (DLL) and Divisor Latch High (DLH)

DLL and DLH store the 16-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most significant part of the divisor; DLL stores the least significant part of the divisor.

2.3.7 Scratch Pad Register (SPR)

SPR is used—as the name implies—as a scratch pad, a temporary memory location for the host to store an 8-bit data byte.

3. UART to Bluetooth interface

This application note will show two examples of hardware interface between products incorporating Bluetooth wireless technology and a UART. One example shows an interface between an off-the-shelf Bluetooth module and a UART, and the other example shows an interface between a Bluetooth chip set made by Philips Semiconductors and a UART.

3.1 Off-the-shelf Bluetooth module

This example shows a Bluetooth Serial Port Plug from Free2Move, but there are other companies that make and sell similar modules. The F2M01 integrates a Bluetooth module in very dense packing, which enables replacement of the most commonly used serial port cable. A class 1 Bluetooth is used, offering a nominal range of approximately 100 meter. No external drivers are needed to use the plug. A user-friendly Windows application is included to program the communication modes as well as the baud rate, and enables hardware flow control using $\overline{RTS}/\overline{CTS}$ signals. Figure 3 shows the interface to an existing serial port 9-pin connector.

The F2M01 needs to be powered by an external power source of 4 to 5 V_{DC} through a DC connector or via pin 9 of the RS-232 DSUB connector. The module can be connected to any RS-232 serial port interface such as a PC serial port, or to any RS-232 9-pin DSUB connector. The module offers data rates up to 115.2 kbit/s.

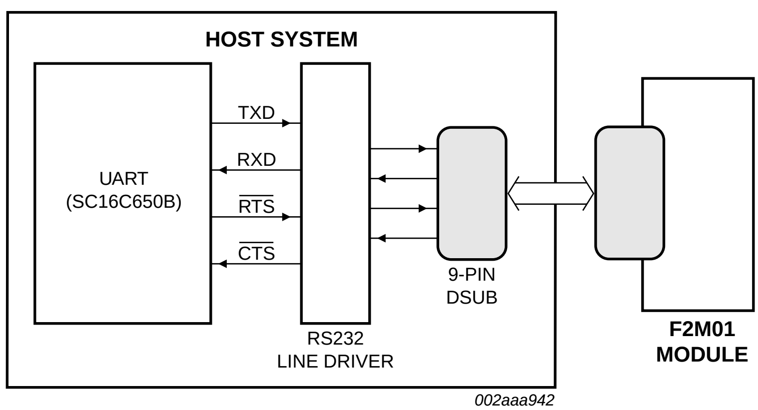

Fig 3. Bluetooth interface to system's serial port.

3.2 Bluetooth chip set

Philips PCF87752 is the most highly integrated single-chip baseband solution designed for Bluetooth applications. Known as ‘Blueberry DATA’, it contains everything required to provide applications with complete baseband functionality for Bluetooth wireless technology. The IC is the optimal choice for Bluetooth integration in hosted and some embedded applications, such as mobile phones and computing devices.

The central processor in Blueberry DATA is an embedded 32-bit ARM7DMI RISC microcontroller, offering very low power consumption per MHz. Several industry-standard I/O interfaces are included, for example: USB, UART, I^{2}C, PCM/IOM. These enable the device to communicate with a wide range of devices.

Although Blueberry offers several I/O interfaces, the simpler transport protocol of the UART results in much less software overhead, and a more cost-effective hardware solution. Figure 4 shows such an interface, and Figure 5 shows a detailed interface of a UART to host controller.

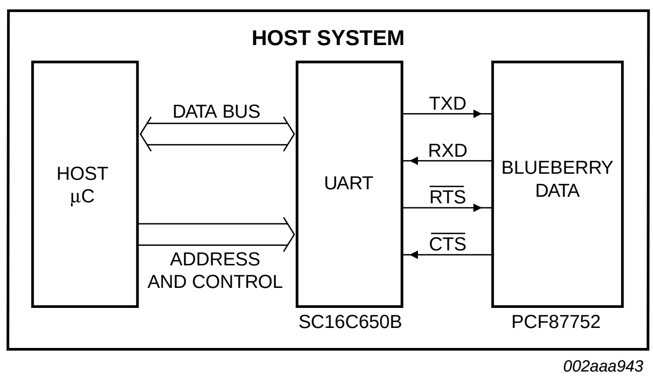

Fig 4. Bluetooth interface to system.

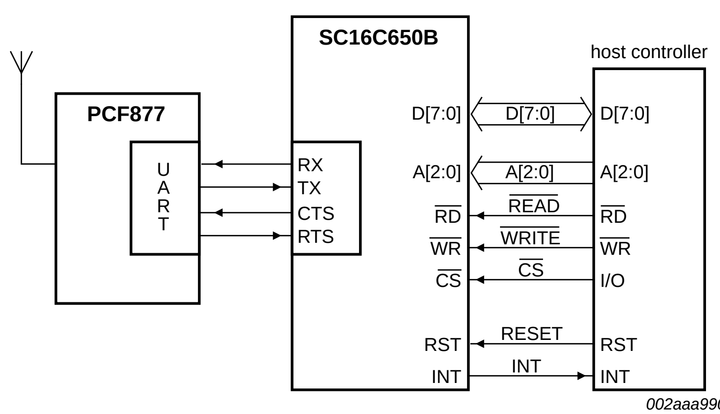

Fig 5. Interface with Bluetooth

4. Conclusion

Bluetooth is an emerging wireless technology that is fast, convenient, and can easily be integrated into mobile and battery operated applications. Bluetooth semiconductors companies offer many versatile host interfaces to allow to this integration to be glueless and requires no external component, but a UART provides a simple, proven and fast interface. To our knowledge, most of the Bluetooth chip suppliers include a UART in their host-controller interface. With a growing trend to faster data rates, Philips' UART solutions provide the speed, simplicity and flexibility to interface with today's and tomorrow's Bluetooth solutions.

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

More information on Philips' Bluetooth solutions can be found at: http://www.semiconductors.philips.com/technologies/bluetooth/.

More information specific to Bluetooth (manufacturers, specification, etc.) can be found at: http://www.bluetooth.com and https://www.bluetooth.org/.

5. 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.

6. Trademarks

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.

7. Contents

Date of release: 11 August 2004
Document order number: 9397 750 13929

Published in the U.S.A.