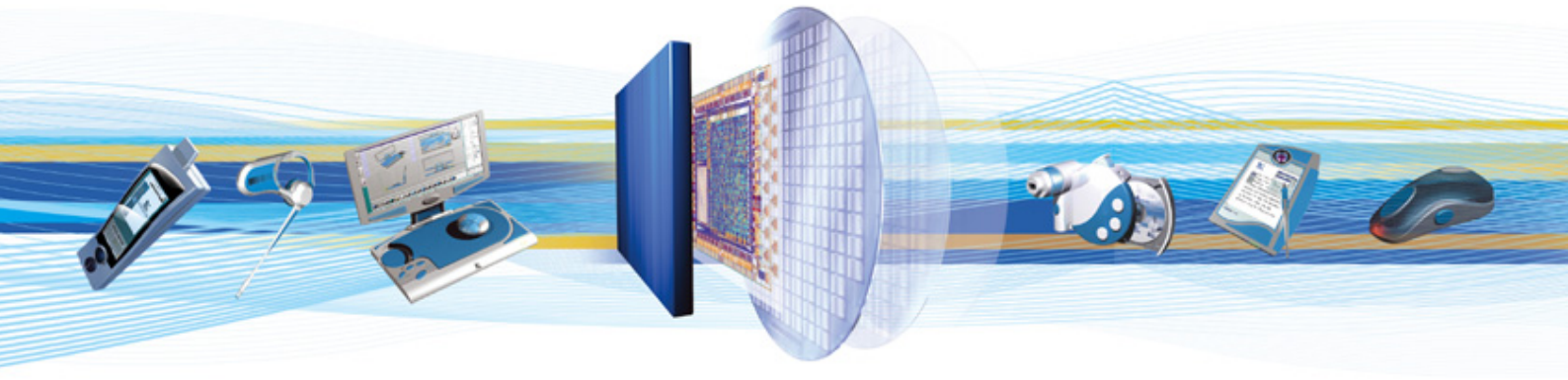
BlueCore™

UART Baud Rate Adaptation

June 2005



CSR

Churchill House
Cambridge Business Park
Cowley Road
Cambridge CB4 0WZ
United Kingdom

Registered in England 3665875

Tel: +44 (0)1223 692000
Fax: +44 (0)1223 692001
www.csr.com

Contents

1 Introduction

Firmware build HCIStack1.1v17.6.3 introduced support to allow UART-based host transports to work without initially knowing the host's baud rate. The firmware examines the host's initial UART traffic and configures its UART to match.

This mechanism works well with BlueCore™ Serial Protocol (BCSP). If the host uses a modified initialisation sequence, it can also be made to work with H4.

The mechanism has been designed to work where the host uses a baud rate between 38.4 kbaud and 691.2 kbaud, although the range can be wider.

Where a dual boot mechanism is used (which is often the case when using a BlueCore2-ROM part), the chip can be configured to use a specific baud rate from the second boot.

2 UART Baud Rate Adaptation

2.1 Basic Operation

In firmware builds before HCIStack1.1v17.6.3, when the firmware initialised a UART-based host transport it read the required baud rate from a PS Key and configured the UART accordingly. From build HCIStack1.1v17.6.3, if the PS Key's required baud rate is zero (the default for the build), then a new UART baud rate adaptation mechanism is invoked. (If the PS Key's baud rate is non-zero, then the value is loaded into the UART, as in previous builds.)

The UART baud rate adaptation mechanism starts by treating the UART's RxData line as a simple PIO input, and it records the host's initial UART traffic in a buffer. The mechanism then examines the recorded data stream to determine the times between signal level transitions and, from this, derives the baud rate in use. The mechanism then programs the UART to match the measured baud rate.

The amount of initial data needed for analysis depends on:

- The data itself (a random mix of byte values is best)
- The actual baud rate

Fortunately, a BCSP Link Establishment Sync message (the first message on a BCSP connection) is a reasonably good data set for the analysis to complete, though at high baud rates several Sync messages may be necessary.

This mechanism must complete before any normal UART traffic is passed to or from the host transport driver. For example, BCSP Link Establishment messages are neither transmitted nor responded to until the UART's baud rate has been set.

The mechanism is independent of the UART's other configuration controls (parity, number of stop bits, etc.). These are normally set to suit the selected host transport.

The mechanism has been designed to operate between 38.4 kbaud and 691.2 kbaud, though testing has concentrated on operation between 56 kbaud and 115.2 kbaud. Implementers are therefore advised to choose a target initial baud rate within these narrower limits. Otherwise, they are advised to conduct their own rigorous testing program.

The mechanism can theoretically operate between 14.4 kbaud and 1382.4 kbaud, but testing has been very light near these limits.

BlueCore's baud rate detection algorithm currently assumes that the gap between UART bytes is an integer multiple of the bit period. This can present problems when using irregular inter-byte spacing.

2.2 Operating with an Unknown System Clock Frequency

The mechanism can operate where the system's clock frequency is unknown at boot, as described in [ROMBOOT]. This can cause BlueCore to run more slowly than normal (typically around 20% slower, though it can run as slow as half speed). Consequently, the range of acceptable baud rates is shifted down pro-rata.

When BlueCore adapts to an unknown system clock frequency, the actual rate at which it runs depends on the physical characteristics of the chip. For example, one BlueCore chip may lock onto the system clock and run 20% slower than normal. However, due to normal manufacturing process variation, another chip may run 30% slower in identical conditions. It is because of this variation that the advice is given above that implementers conduct their own testing if the target initial baud rate is outside the core range of 56 to 115.2 kbaud.

Operation with an unknown system clock implies dual-boot initialisation of BlueCore. A reasonable approach is for the host to use a safe baud rate after the first boot (e.g., 115.2 kbaud), then to set a new baud rate in the PS Key store (along with the device's Bluetooth address, actual system clock frequency, etc.), and then to use the new baud rate from when the chip is warm-rebooted.

2.3 Operation with BCSP

When a BCSP link starts operation, the two devices send Link Establishment Sync messages to each other. These are unreliable and are sent repeatedly until they are acknowledged. This pattern fits well with the baud rate adaptation mechanism: the host's first few Sync messages are used by BlueCore to match its baud rate to the host; then the host's next Sync message is accepted and responded to. The only consequence is a small extra boot delay; Sync messages are normally sent every 250ms.

To speed up connection, it may help to send Sync messages more frequently, e.g., once every 50ms, but it is inadvisable to send them more frequently than this.

2.4 Operation with H4

The H4 protocol expects UART communication to be perfect. The baud rate adaptation mechanism analyses and discards the host's first characters, so it appears to be unworkable with H4. However, H4's own error recovery mechanism can be used:

- The host sends characters to BlueCore. It is best if these do not contain legal H4 message start values (`0x01` to `0x04`). A continuous stream of a single character is also not suitable, as the adaptation mechanism requires a mix of byte values. Incongruously, BCSP Link Establishment Sync messages work well. As with BCSP, it is best not to flood the link with traffic at this initial stage, as it can fill the firmware's memory.
- The baud rate adaptation mechanism collects a block of these characters and sets its UART to match the host's baud rate.
- The host continues to send characters. These are passed to the BlueCore H4 driver, which recognises them as not forming legal H4 messages, so it invokes its H4 recovery mechanism: it blocks normal HCI traffic and sends an HCI Hardware_Error event to the host with error code `0xfe`.
- The host receives the error code, stops sending non-H4 traffic and sends an H4 HCI Reset command.
- The BlueCore H4 driver recognises the HCI Reset command, after which normal HCI traffic can flow.

Although this initialisation sequence appears logical, at the time of writing this document, it has not been tested.

Document References

Document ID	Document Title	CSR Reference
[BT1.1]	Specification of the Bluetooth System, Version 1.1, volume 1, Core, 22 February 2001	n/a
[ROMBOOT]	Booting BlueCore ROM	bcore-me-014P

Terms and Definitions

BCSP	BlueCore Serial Protocol
BlueCore™	Group term for CSR's range of Bluetooth wireless technology chips
BlueCore2-ROM	CSR Bluetooth chip
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
H4	UART-based HCI transport; described in [BT1.1]
Persistent Store	Storage of BlueCore's configuration values in non-volatile memory
PIO	Programmable Input Output
PS Key	Persistent Store Key
UART	Universal Asynchronous Receiver Transmitter

Document History

Revision	Date	History
a	12 Dec 03	Original publication of this document.
b	1 Mar 04	Updated firmware reference.
c	6 Jul 04	Updated formatting, minor corrections.
d	27 Jun 05	Updated section 2.1 to include a reference to baud rate detection problems when using irregular inter-byte spacing. Incorporates minor formatting updates.

BlueCore™

UART Baud Rate Adaptation

bcore-me-019Pd

June 2005