BTM44X ENHANCED DATA MODULE: LATENCY OPTIMIZATION

Application Note *v1.0*

INTRODUCTION

The BTM44X Enhanced Data module embeds a full Bluetooth stack up to and including the highest RFCOMM layer which is used in a Serial Port Profile (SPP) connection.

RFCOMM uses L2CAP and many other profiles. For example, a HID uses L2CAP directly.

RFCOMM is a streaming protocol while L2CAP is packet based.

There are many use cases (Serial Port Profile or HID) where getting the information from one end to the other should happen with minimum delay. For example, a HID reports from keyboards and mice.

This application note explains how to configure the BTM44X so that the best latency (least transit delay) is achieved.

DATA FLOW ANALYSIS

Once a Bluetooth connection is established, data from a host flows through the BTM44X, which must first get into the module via the UART interface. Then the application inside the module bridges it to the RF interface so that it goes 'over the air'.

Latency, for the purpose of this application note, is defined as the time it takes for a single byte to get from the host to a peer device and then on to its host over a Bluetooth connection.

If the host and the peer are physically connected via a UART interface, then the latency is defined solely by the baud rate and the interrupt response time at the receiving end. Assuming a baud rate of BAUD with no parity and one stop bit, an eight bit byte will require 10 clock periods to get to the other side's receive buffer. With that, the latency is 10/BAUD seconds. At 115200 baud lasting roughly 87 microseconds, plus another 10 microseconds for the interrupt response, the total is approximately 100µsec. This is referred to Delay A in this application note.

When the same data is sent over a Bluetooth connection using the BTM44X Serial to Bluetooth module, then (in addition to Delay A so that the byte ends up in the module) there is a delay associated with the underlying operating system throwing an event into the Laird application running in the module (Delay B).

The application in turn has to queue the byte in a holding buffer in case more data from the UART is forthcoming. Since the application cannot hold the data in the buffer indefinitely, it starts a 'flush' timer (Delay C).

When the timer expires, whatever is in the buffer is presented to the RFCOMM (or L2CAP) layer for forward transmission via the Bluetooth baseband.The baseband radio itself may be busy doing other tasks, such as listening for incoming connections or inquiries from other BT devices. Connection data must wait for the radio to become free before that data can be sent (Delay D).

The 'on-air' Bluetooth protocol is packet-based in principle, where each packet is synchronized to a 1.25 milliseconds clock; this introduces another delay (Delay E).

To aggravate the situation more, if power saving has been enabled using Sniff Mode, then the first data packet must be synchronized to the beginning of the Sniff Window interval (Delay F).

Finally, if the RF connection quality is bad, retries occur until a packet is acknowledged. These retries occur every 1.25msec (Delay E * n). In the worst case scenario where Sniff is enabled, the Sniff Attempt and Timeout settings ensure that the retries occur at multiples of 1.25msec and not the Sniff Window interval.

For the purpose of this analysis, the time of transmission 'on-air' is deemed negligible. At the speed of light, and for a wireless range of 100 meters, this corresponds to a time of about 1/3 of a microsecond.

On the receiving Bluetooth device, if it is also a Bluetooth serial module, there is a small negligible delay (Delay G) for the data to be copied into the UART transmit buffer. Finally, the transmission delay (Delay H) over the UART line at a baud rate configured in that module may be different from the baud rate set in the transmitting module.

CONFIGURING FOR OPTIMIZED LATENCY

As described in the previous section, the latency for a character to be sent from one end to the other is the sum of WORST CASE delays A to H. This section explains how to configure the BTM44X module so that each delay (and therefore the sum) is appropriately minimized.

LATENCY = Sum of (Worst case delays from A to H)

If the connection quality is not good, retries occur that increase the latency by 1.25*N milliseconds (where N is the number of retries for each packet).

Delay A: UART Serialization of Byte (Receive)

This delay is minimized by setting the baud rate to the highest possible setting; for the BTM44X the highest baud is 921600. Minimize the delay by setting S Register 240 (MP mode) or S Reg 520/521 (AT mode).

Note: When setting this register, storing to non-volatile memory is automatic and CMD_STORE_SREG (MP mode) or AT&W (AT mode) is not required.

Delay B: Operating System Event Delay

This delay is minimized by setting S Register 84 (MP mode) or S Register 9084 (AT mode) to three so the lowest level UART buffers are polled every 1 millisecond. If this register is left to the default value of zero, the polling interval will be 100 milliseconds.

A CMD_STORE_SREG (MP mode) or AT&W (AT mode) is required to save to non-volatile memory so that on subsequent incarnations the UART is configured accordingly.

Note: Setting this register for best latency will have a detrimental impact on throughput as the processor will be spending more time servicing the UART polling routine.

Delay C: UART to RFCOMM/L2CAP Bridging Delay

This delay is adjusted using S Register 80 (MP mode) or S Register 9080 (AT mode) to the minimum value possible so that small chunks of data sit in a holding buffer for the shortest amount of time.

Note: A CMD_STORE_SREG (MP mode) or AT&W (AT mode) is required to save to non-volatile memory so that on subsequent incarnations the UART is configured accordingly.

Delay D: Baseband Queuing Delay

When in a connection, the baseband radio is timeshared with any ongoing operation associated with listening for incoming connections and inquiries.

If you know the module always makes outgoing connections, never accepts incoming connections, and is only discoverable at user determined times, then set S Registers 4 & 5 (MP mode) or 9004 & 9005 (AT mode) to zero.

If it is necessary to be connectable while in a connection, then reduce the paging duty cycle to the minimum by setting S Registers 9 & 10 (MP mode) or 9009 & 9010 (AT mode) to 2560 and 12 respectively. This means that the module spends only 12 milliseconds in every 2560 milliseconds listening for a connection.

Delay E: Baseband Slot Timing Delay

This potential delay of 1.25 milliseconds is a feature of the architecture of Bluetooth communications and cannot be adjusted.

Delay F: Sniff Window Interval

This parameter conflicts with latency. The sniff operation needs to be as aggressive as possible so that power consumption is minimized, but the requirements of latency push it in the opposite direction.

The sniff parameters are adjusted using S registers 73 to 76 inclusive in MP mode (9073 to 9076 or 561 to 564 in AT mode). To improve latency:

- Set S Register 75 & 76 as small as possible (to define the delay)
- For latency, set 73 as small as possible.

Note: A CMD_STORE_SREG (MP mode) or AT&W (AT mode) is required to save to non-volatile memory so that on subsequent incarnations the UART is configured accordingly.

Delay G: Incoming Data to UART Delay

This is a feature of the architecture of the underlying Bluetooth stack and cannot be adjusted

Delay H: UART Serialization of Byte (Transmit)

Only relevant if the receiving Bluetooth device is also a serial module.

This can be minimized by setting the baud rate to the highest possible number by setting S Register 240 (MP mode) or S Reg 520/521 (AT mode). The highest baud rate for the BTM44X is 921600.

Note: When setting this register a store to non-volatile memory is automatic and CMD_STORE_SREG (MP mode) or AT&W (AT mode) is not required.

Revision History

Revision	Date	Description	Approved By
1.0	28 June 2012	Initial Release	Jonathan Kaye