Application Note

AN2337/D
Rev. 0, 2/2003

Bluetooth Park Mode

Contents

1 Introduction

This document contributes to the discussion about the future of Bluetooth Park Mode.

Motorola implemented Park Mode in our first and second generation Bluetooth Platforms: P1.0 and P1.5. Motorola has tested P1.0 and P1.5 against many other vendors. As one of the Bluetooth front-runners, Motorola has been heavily involved in Park interoperability testing and the subsequent discussions on Park interoperability problems which revealed a lot of the inherent Park Mode interoperability problems. These problems are not only recognized internally in Motorola, but also commonly known in the Bluetooth community.

A white paper proposal that contains Park Mode parameter guidelines was written, so some progress has been made, but to improve the Park Mode specification, a lot of errata work is still needed. As this application note will show, a good alternative to Park Mode already exists in the Bluetooth specification. This alternative solution uses Sniff Mode with the master transmitting NULL packets. It is preferable to use this alternative solution for several reasons including, no known interoperability problems, and a less complex system, which mean lower cost.

2 Description of Park Mode

Slave devices can use Park Mode when there is no current need for active participation in the piconet, but the slave device wants to remain synchronized. The parked slave is not an active member of the piconet and can, therefore, neither receive nor transmit point-to-point L2CAP data. The unreliable broadcast L2CAP packets are the only data traffic received. If the parked slave device wants to rejoin the piconet as an active member, it must be unparked.

The parked slave only periodically listens to the traffic on the piconet. To allow the piconet synchronization, the broadcast L2CAP traffic, and the unpark negotiation, the master supports a beacon channel. On this channel, traffic is going on at regular intervals, where the interval can be up to 40.96s. When not listening, the slave device can enter low power mode which reduces the power consumption.

When entering Park Mode, the slave device loses its active member address which allows it to be used by the master to connect to or unpark another slave. Using Park Mode will, therefore, increase the theoretical number of possible slaves from 7 to 4096.

To summarize, the main purposes of Park Mode are to:

- Reduce power consumption because the devices in Park Mode are only awake periodically.
- Allow more than seven slaves to participate in a single piconet.

As mentioned in the introduction of this application note, the current Park Mode specification has some inherent interoperability problems. In addition to this, the specification does not tell how to handle common situations such as the order of the various substates in the beacon channel, and sequence numbering for broadcast traffic. Finally, the number of Park Mode parameters makes setting ranges very difficult. Therefore, Park Mode needs many errata to be properly specified.

In terms of quality of service (QoS), Park Mode has two inherent problems that make it unusable. First, unpark can take up to “supervision timeout” which is set to 20s per default. Second, scheduling of traffic to more than the seven active slave devices simultaneously requires indirect scheduling for park/unpark. This makes the scheduling unpredictable and can easily result in QoS violation and even data packet starvation.

Finally, it is questionable if a device needs more than seven slaves. Support for additional slaves will add to the requirement for data memory. As no latency guarantees on unpark are given, it could be as good just to disconnect and then, when the device is to be addressed again, reconnect to it.

3 Alternatives

The alternative solution to Park Mode is using Sniff Mode with the master transmitting NULL packets instead of using POLL packet transmission. The advantage of sending NULL packets is that the slave does not have to respond, which means it can save power. The slave device only has to respond to the master transmitted packet when an update of the supervision timeout on the master side is required. Think of Sniff Mode as a point-to-point link with a lower duty cycle instead of as a device in active mode. Otherwise, there are no differences between the two modes as the device is still an active member of the piconet when in Sniff Mode.

As far as power consumption is concerned, the Sniff Mode approach is indistinguishable from Park Mode. Actually, Sniff Mode uses a little less power since regular LMP negotiations are not required, and the beacon channel signaling requires more bandwidth than the NULL packet transmissions.

The Sniff Mode specification, which is very simple compared to Park Mode, has no known interoperability problems.

Regarding QoS, Sniff Mode is far better than Park Mode since the latency for the next possible L2CAP traffic is given by the specified sniff interval. Remember, the slave does not have to respond if there is nothing to transmit, so the sniff interval can be set to a reasonably low value without violating the desire to save power. In addition, the Sniff Mode attempt windows can be used as a means of supporting guaranteed bandwidth allocation.

4 Conclusion

The Park Mode functionality can easily be substituted by Sniff Mode. The only real concern is the number of possible slaves simultaneously participating in the piconet.

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

NOTES

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.; Silicon Harbour
Centre, 2 Dai King Street, Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://www.motorola.com/semiconductors

MOTOROLA

AN2337/D

Preliminary