Application Note

94001481001
Rev. 1.1, 2/2003

Motorola Bluetooth File System, Overview

MOTOROLA intelligence everywhere™ | digital dna™

Contents

1 Introduction

This document provides an overview of the Motorola Bluetooth file system and presents three different approaches—host based general, host based one file, and embedded file system. The first two approaches enable a solution without external EEPROM.

The three approaches are described in detail in the following application notes:

- *Motorola Bluetooth File System, Host Based General Application Note* (document number 94001481003)
- *Motorola Bluetooth File System, Host Based One File Application Note* (document number 94001481004)
- *Motorola Bluetooth File System, Embedded File System Application Note* (document number 94001481002)

2 Overview

The host interface is a normal serial UART interface, where all communication from the application, upper stack and file system passes through.

For more detailed explanations, see the following documents:

- *MC71000 Bluetooth Baseband Controller data sheet* (document number MC71000TB/D)
- *MC72000 Integrated Bluetooth Radio data sheet* (document number MC72000TB/D)
- *MC71000/MC72000 Wake-up, Reset and Host Clock Request Sequences application note* (document number AN2340/D, revision 0, 9/2002 or newer).

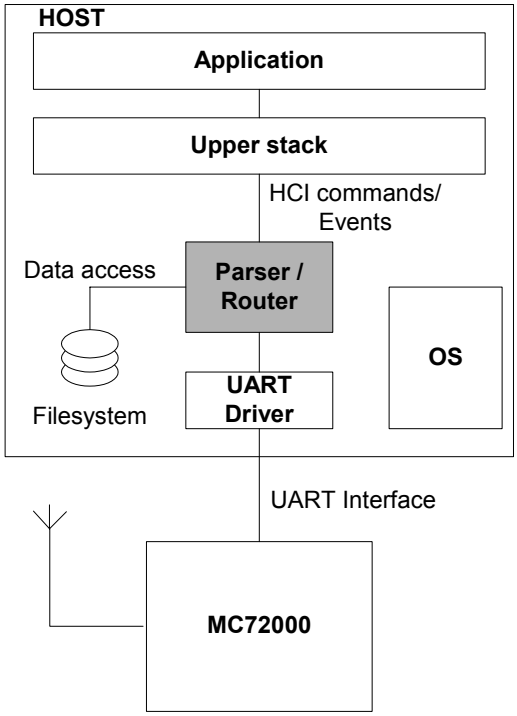

Figure 1. Overview

3 UART Interface Telegrams

The serial UART interface transports all types of HCI telegrams. This includes the following:

- HCI ACL: Data transport—typically a file to/from another unit.
- HCI SCO: Audio transport—typically sound to/from another unit.
- HCI Commands: Are sent to the MC71000/MC72000 ordering it to do something. This will always result in one or more HCI events returned.
- HCI Events: Are sent from MC71000/MC72000 to the host and are a direct response to an HCI command or a spontaneous reaction to something happening in/around the MC71000/MC72000.

4 UART Interface Hardware

The UART driver uses the following signals:

- GND for common ground reference point.
- TxD/RxD for serial data.
- RTS/CTS lines as normal hardware handshake (optional).
- DTR/DSR lines as sleep mode request/grand (optional).

However, the interface can be configured to be a 2, 4, or 6 wire interface. For more details, see *MC71000/MC72000 Wake-up, Reset, and Host Clock Request Sequences Application Note* (document number AN2340/D, revision 0, 9/2002 or newer).

5 UART Driver

The UART driver is highly dependent on the hardware and the operating system of the host system. However, it must conform to the following rules:

1. Transmit and receive at 9600 baud, 8 Bit, 1 StopBit, NoParity at startup.
2. Ability to change baud rate, if needed in this application.
3. Support handshake on RTS/CTS lines, if needed in this application. For more details, refer to *MC72000 Wake-up, Reset, and Host Clock Request Sequences Application Note* (document number AN2340/D, revision 0, 9/2002 or newer).
4. Support sleep mode, if needed in this application. For more details, refer to *MC72000 Wake-up, Reset, and Host Clock Request Sequences Application Note* (document number AN2340/D, revision 0, 9/2002 or newer).
5. A delay of more than 300 mS between characters in a command will result in a “Hardware Error” event returned, telling to retransmit the last telegram. This will impact general speed and should be avoided.

6 Parser/Router

There are three possible ways to implement the file system. Each has great impact on the way the parser/router part must be implemented, as well as cost and hardware. For details, see the following sections.

6.1 General File System on Host

This saves the EEPROM and can use an existing file system on the host.

The **router** has to do the following things:

1. Sort all commands/events/ACL/SCO and send to correct receiver.
2. Send host connect after reset.
3. Change baud rate after host connect, if desired.
4. Resend telegrams on “Hardware error” event received, if desired.
5. Keep track of number of pending commands.

The **parser** has to do the following things:

1. Translate received file system events into local file system calls.
2. Send correct command, with data, on received file system events.
3. No action necessary on “Command Complete” events.

Some way of reprogramming the files on the host has to be considered.

For more details, see *Motorola Bluetooth File System, Host Based General Application Note* (document number 94001481003).

6.2 One-File File System Downloaded to RAM in MC71000/MC72000

This will download an image of all files to RAM in MC71000/MC72000. This saves the EEPROM, and no parser/router has to be present when running application and upper stack code.

The download of this image is done once during initialization of the MC71000/MC72000.

This functionality will be available from ROM 2.1.

No EEPROM is needed, but the host must have initialization software that can make host connect and send one file after all resets. Furthermore, the host must store link keys, if desired, and send those after each reset.

For more details, see *Motorola Bluetooth File System, Host Based One File Application Note* (document number 94001481004).

6.3 File System in EEPROM on MC71000/MC72000

An EEPROM containing the files is connected to the MC71000/MC72000. The parser/router part does not exist in this configuration and the upper stack is directly connected to the UART driver.

Some way of reprogramming the EEPROM normally must be considered. This can be done through JTAG with a special program, or directly through UART interface with a parser program.

This solution will add the cost of an EEPROM but will keep host software simple.

For more details, see *Motorola Bluetooth File System, Embedded File System application note* (document number 94001481002).

7 Files

To make the MC71000/MC72000 as flexible as possible, the following files have been defined.

Table 1. Existing Files

Filename	Name	Description
1000	Boot image	This file is the boot program that controls startup.
2000	Patch	This file contains the patch code. Enabling new features or correcting errors in existing functions in ROM or RAM.
3000	OEM application	The application file contains the customer specific application.
3500	Headset config.	Configuration for a headset application
3600	UART config.	Configuration for a UART based application
3fff	Production test code	Code specific for testing the MC71000/MC72000 and the radio module in a test environment. For details, see *Vendor-Specific HCI Reference (public)* (document number 79000001800, revision 1.0, 2002.09.02, or newer).
4000	Baseband NVM	NVM parameters for the baseband, that is, Bluetooth address, link keys, country code and sleep mode setup, etc.
4100	Radio parameters/patch	Radio register values in different setups and software patch code for the radio.
4200	Production data	Hardware name string, serial number, revision, etc.
5500	ROM image	The ROM image file is a complete copy of complete EEPROM.
6500	RAM image	The image of a file system to be accessed from RAM.

Not all of the files will be present in all configurations. For more details, see the tree detailed documents listed in the Introduction of this overview application note.

All files are created by tools or provided by Motorola.

8 File Loading

The files will typically be loaded in the following sequence for a host-based general file system using ROM 2.0, with a UART application as shown in Figure 2.

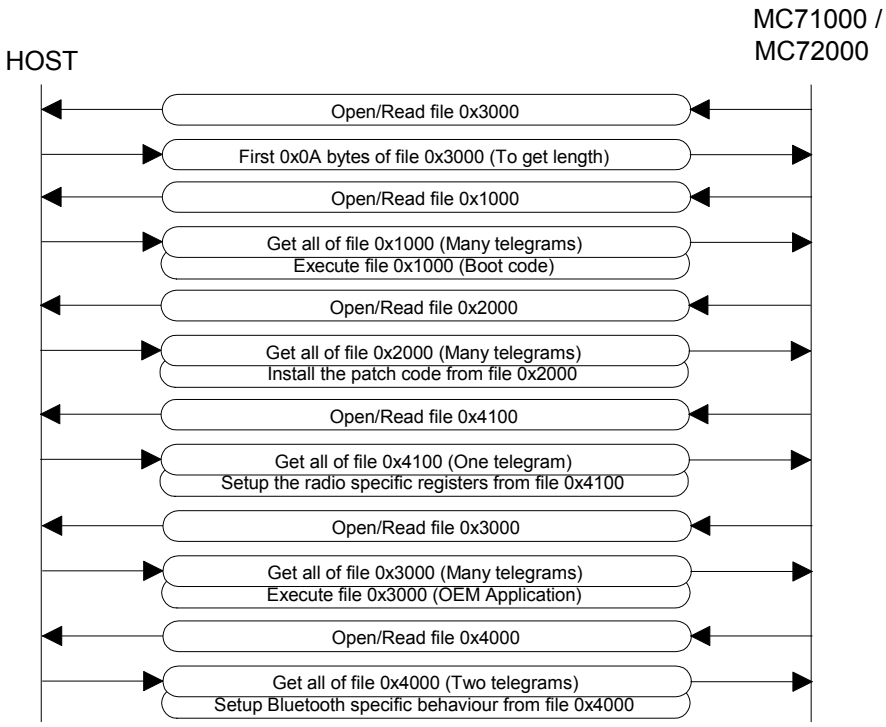

Figure 2. Host Based General—File Load Sequence

For more details, please refer to *Motorola Bluetooth File System, Host Based General application note* (document number 94001481003).

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

94001481001

Preliminary