

U S E R M A N U A L

Copyright (C) 2003-2010, Marvell International Ltd.

This software file (the "File") is distributed by Marvell International Ltd. under the terms of the GNU General Public License Version 2, June 1991 (the "License"). You may use, redistribute and/or modify this File in accordance with the terms and conditions of the License, a copy of which is available along with the File in the gpl.txt file or by writing to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 or on the worldwide web at <http://www.gnu.org/licenses/gpl.txt>.

THE FILE IS DISTRIBUTED AS-IS, WITHOUT WARRANTY OF ANY KIND, AND THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. The License provides additional details about this warranty disclaimer.

1) FOR DRIVER BUILD

Goto source code directory `bt_src/`.  
`make [clean] build`  
 The driver binary can be found in `../bin_xxxx_bt` directory.

2) FOR DRIVER INSTALL

- a) Copy `sd8790.bin` | `sd8787.bin` | ... to `/lib/firmware/mrvl/` directory, create the directory if it doesn't exist.
- b) Install bluetooth driver,  
`insmod bt8688.ko` | `bt8790.ko` | `mbt8787.ko` | ...
- c) Uninstall bluetooth driver and sdio bus driver,  
`hciconfig hciX down`  
`rmmmod bt8xxx` | `mbt8xxx`

3) `cat /proc/mbt/hcix/status`

This command is used to get driver status.

4) `cat /proc/mbt/hcix/config`

This command is used to get the current driver settings.

5) `proc` commands to config bluetooth parameters

`drvdbg=[n]`

- This command is used to set the bit masks of driver debug message control.

bit 0:	MSG	PRINTM(MSG,...)
bit 1:	FATAL	PRINTM(FATAL,...)
bit 2:	ERROR	PRINTM(ERROR,...)
bit 3:	DATA	PRINTM(DATA,...)
bit 4:	CMD	PRINTM(CMD,...)
bit 5:	EVENT	PRINTM(EVENT,...)
bit 6:	INTR	PRINTM(INTR,...)
...		
bit 16:	DAT_D	PRINTM(DAT_D,...), DBG_HEXDUMP(DAT_D,...)
bit 17:	CMD_D	PRINTM(CMD_D,...), DBG_HEXDUMP(CMD_D,...)
...		
bit 28:	ENTRY	PRINTM(ENTRY,...), ENTER(), LEAVE()
bit 29:	WARN	PRINTM(WARN,...)
bit 30:	INFO	PRINTM(INFO,...)

Usage:

`echo "drvdbg=0x7" > /proc/mbt/hcix/config`

`#enable MSG, FATAL, ERROR messages`

`gpio_gap=[n]`

This command is used to configure the host sleep parameters.

bit 8:0 -- Gap  
 bit 16:8 -- GPIO  
 where GPIO is the pin number of GPIO used to wakeup the host. It could be any valid GPIO pin# (e.g. 0-7) or 0xff (Interface, e.g. SDIO will be used instead).  
 where Gap is the gap in milli seconds between wakeup signal and wakeup event or 0xff for special setting.

README-bt.txt

Usage:

```
echo "gpio_gap=0xff80" > /proc/mbt/hcix/config      # use Interface (e.g. SDIO)
echo "hscfgcmd=1" > /proc/mbt/hcix/config            # gap = 0x80

echo "gpio_gap=0x03ff" > /proc/mbt/hcix/config      # use gpio 3
echo "hscfgcmd=1" > /proc/mbt/hcix/config            # and special host sleep mode
```

psmode=[n]

This command is used to enable/disable auto sleep mode

where the option is:

```
1      -- Enable auto sleep mode
0      -- Disable auto sleep mode
```

Usage:

```
echo "psmode=1" > /proc/mbt/hcix/config              #enable power save mode
echo "pscmd=1" > /proc/mbt/hcix/config

echo "psmode=0" > /proc/mbt/hcix/config              #disable power save mode
echo "pscmd=1" > /proc/mbt/hcix/config
```

6) Use hcitool to issue raw hci command, refer to hcitool manual

Usage: Hcitool cmd <ogf> <ocf> [Parameters]

1. Interface Control Command

```
hcitool cmd 0x3f 0x5b 0xf5 0x01 0x00    --Enable All interface
hcitool cmd 0x3f 0x5b 0xf5 0x01 0x01    --Enable Wlan interface
hcitool cmd 0x3f 0x5b 0xf5 0x01 0x02    --Enable BT interface
hcitool cmd 0x3f 0x5b 0xf5 0x00 0x00    --Disable All interface
hcitool cmd 0x3f 0x5b 0xf5 0x00 0x01    --Disable Wlan interface
hcitool cmd 0x3f 0x5b 0xf5 0x00 0x02    --Disable BT interface
```

7) cat /proc/mbt/hcix/debug

This command is used to get driver debug parameters.

8) proc command to config debug parameters

sdcmd52rw=<func> <reg> [data]

This command is used to read/write a controller register in Secure Digital I/O Interfaces.

func: The function number to use (0-7)

reg: The address of the register

data: The value to write, read if the value is absent

For SDIO MMC driver, only function 0 and BT function (2/3) access is allowed.

And there is a limitation for function 0 write, only vendor specific CCCR registers (0xf0-0xff) are permitted.

Usage:

```
echo "sdcmd52rw= 2 3 0xf" > /proc/mbt/hcix/debug    # write 0xf to func 2 address 3
echo "sdcmd52rw= 0 4" > /proc/mbt/hcix/debug         # read func 0 address 4
```

=====