1. **ec80fad0-5659-41d3-ae3a-28bf4339809e**  
   This documentation is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA For more details see the file COPYING in the source distribution of Linux.
   1. This documentation is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA For more details see the file COPYING in the source distribution of Linux.
   2. Как 1.1
   3. This documentation is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA For more details see the file COPYING in the source distribution of Linux.
   4. Как 1.3
   5. Как 1.3
   6. Как 1.1
   7. This documentation is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation. This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this documentation; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA For more details see the file COPYING in the source distribution of Linux.
   8. Как 1.1
   9. Как 1.3
   10. Как 1.1
   11. Как 1.3
   12. Как 1.3
   13. Как 1.3
   14. Как 1.3
   15. Как 1.1
   16. Как 1.3
   17. Как 1.3
   18. Как 1.1
   19. Как 1.3
   20. Как 1.1
   21. Как 1.3
   22. Как 1.3
   23. This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.
   24. Как 1.23
   25. This documentation is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. For more details see the file COPYING in the source distribution of Linux.
2. **7eec4b69-b4da-4bf3-9d0e-74a1af7a7786 2 раза, проигнорировано. М.б. техническая ошибка импорта.**  
   These books attempt to give a description of the various subsystems that play a role in 802.11 wireless networking in Linux. Since these books are for kernel developers they attempts to document the structurescand functions used in the kernel as well as giving achigher-level overview.  
   ...  
   STA information lifetime rules  
   Aggregation
3. **d4065f0e-8019-48a2-b63c-067e454c84da**  
   This chapter contains the autogenerated documentation of the…
   1. This chapter contains the autogenerated documentation of the internal functions.
   2. This chapter contains the autogenerated documentation of the kernel API functions which are exported.
   3. chapter entitled &quot;GNU Free Documentation License&quot
   4. This chapter contains the autogenerated documentation of the structures which are used in the generic IRQ layer.
   5. This chapter contains the autogenerated documentation of the RapidIO subsystem.
   6. This chapter contains the autogenerated documentation of the structures which are used in the Reed-Solomon Library and are relevant for a developer.
   7. This chapter contains the autogenerated documentation of the Reed-Solomon functions which are exported.
4. **cc12f3a0-07a8-4442-9cb5-bade952eb39e**  
   This chapter contains the autogenerated documentation of the structures which are used in the NAND driver and might be relevant for a driver developer. Each struct member has a short description which is marked with an [XXX] identifier. See the chapter &quot;Documentation hints&quot; for an explanation.
   1. This chapter contains the autogenerated documentation of the structures which are used in the NAND driver and might be relevant for a driver developer. Each struct member has a short description which is marked with an [XXX] identifier. See the chapter &quot;Documentation hints&quot; for an explanation.
   2. This chapter contains the autogenerated documentation of the NAND kernel API functions which are exported. Each function has a short description which is marked with an [XXX] identifier. See the chapter &quot;Documentation hints&quot; for an explanation.
   3. This chapter contains the autogenerated documentation of the NAND driver internal functions. Each function has a short description which is marked with an [XXX] identifier. See the chapter &quot;Documentation hints&quot; for an explanation. The functions marked with [DEFAULT] might be relevant for a board driver developer.
5. **0cce7955-9606-4b49-b827-1371909f1266**  
   busgid=NNNNNControls the GID used for the /proc/bus/usb/BBB directories. (Default: 0)
   1. busgid=NNNNNControls the GID used for the /proc/bus/usb/BBB directories. (Default: 0)
   2. busmode=MMMControls the file mode used for the /proc/bus/usb/BBB directories. (Default: 0555)
   3. busuid=NNNNNControls the UID used for the /proc/bus/usb/BBB directories. (Default: 0)
   4. devgid=NNNNNControls the GID used for the /proc/bus/usb/BBB/DDD files. (Default: 0)
   5. devmode=MMMControls the file mode used for the /proc/bus/usb/BBB/DDD files. (Default: 0644)
   6. devuid=NNNNNControls the UID used for the /proc/bus/usb/BBB/DDD files. (Default: 0)
   7. listgid=NNNNNControls the GID used for the /proc/bus/usb/devices and drivers files. (Default: 0)
   8. listmode=MMMControls the file mode used for the /proc/bus/usb/devices and drivers files. (Default: 0444)
   9. listuid=NNNNNControls the UID used for the /proc/bus/usb/devices and drivers files. (Default: 0)
6. **71be5ba8-17b5-4251-a6ad-965a514720b1**
   1. 12x File modification time is not updated by this request.
7. **e46b9eea-dfab-44cd-9478-c5fbbec04463**  
   This function is called whenever the initialization function of a real object which resides on the stack is called.
   1. This function is called whenever the initialization function of a real object which resides on the stack is called.
   2. This function is called whenever the activation function of a real object is called.
   3. This function is called whenever the initialization function of a real object is called.
   4. This function is called whenever the deactivation function of a real object is called.
8. **ca5d405c-94c8-471c-b04e-fbea762b78e3**  
   When the real object is already tracked by debugobjects it is checked, whether the object can be activated. Activating is not allowed for active and destroyed objects. When debugobjects detects an error, then it calls the fixup\_activate function of the object type description structure if provided by the caller. The fixup function can correct the problem before the real activation of the object happens. E.g. it can deactivate an active object in order to prevent damage to the subsystem.
   1. When the real object is tracked by debugobjects it is checked, whether the object can be freed. Free is not allowed for active objects. When debugobjects detects an error, then it calls the fixup\_free function of the object type description structure if provided by the caller. The **fixup** function can correct the problem before the real free of the object happens. E.g. it can **deactivate** an active object in order to prevent damage to the subsystem.
   2. When the real object is tracked by debugobjects it is checked, whether the object can be **destroyed**. Destruction is not allowed for active and destroyed objects. When debugobjects detects an error, then it calls the fixup\_destroy function of the object type description structure if provided by the caller. The fixup function can correct the problem before the real **destruction** of the object happens. E.g. it can **deactivate** an active object in order to prevent damage to the subsystem.
   3. When the real object is already tracked by debugobjects it is checked, whether the object can be **activated**. Activating is not allowed for active and destroyed objects. When debugobjects detects an error, then it calls the fixup\_activate function of the object type description structure if provided by the caller. The fixup function can correct the problem before the real **activation** of the object happens. E.g. it can **deactivate** an active object in order to prevent damage to the subsystem.
   4. When the real object is already tracked by debugobjects it is checked, whether the object can be **initialized**. Initializing is not allowed for active and destroyed objects. When debugobjects detects an error, then it calls the fixup\_init function of the object type description structure if provided by the caller. The fixup function can correct the problem before the real initialization of the object happens. E.g. it can deactivate an active object in order to prevent damage to the subsystem.
   5. Как 7.4
9. **0ebc4e55-09c3-429f-87e6-f4267cfcfbe0**  
   The following people have contributed to this document:
   1. The following people have contributed to the NAND driver:
   2. **3 раза** The following people have contributed to this document:
10. ???? – Кусок нескольких таблиц, побился при импорте. Строка таблицы может генерироваться по шаблону. Несколько десятков строк.  
    Error correction code byte 2 of the lower 256 Bytes of data in this page0x03ECC byte.  
    Проигнорирован, поскольку тяжело восстановить.
11. **8310e538-583d-485b-abfd-aa491af61a9f**  
    These routines disable soft interrupts on the local CPU, and restore them. They are reentrant;
    1. These routines disable soft interrupts on the local CPU, and restore them. They are reentrant;
    2. These routines disable soft interrupts on the local CPU, and restore them. They are reentrant;
12. 24761f84-d434-4527-be68-3d71e3e90657

static struct snd\_pcm\_hardware snd\_mychip\_playback\_hw = {

.info = (SNDRV\_PCM\_INFO\_MMAP |

SNDRV\_PCM\_INFO\_INTERLEAVED |

SNDRV\_PCM\_INFO\_BLOCK\_TRANSFER |

SNDRV\_PCM\_INFO\_MMAP\_VALID),

.formats = SNDRV\_PCM\_FMTBIT\_S16\_LE,

.rates = SNDRV\_PCM\_RATE\_8000\_48000,

.rate\_min = 8000,

.rate\_max = 48000,

.channels\_min = 2,

.channels\_max = 2,

.buffer\_bytes\_max = 32768,

.period\_bytes\_min = 4096,

.period\_bytes\_max = 32768,

.periods\_min = 1,

.periods\_max = 1024,

};

static struct snd\_pcm\_hardware snd\_mychip\_playback\_hw = {

.info = (SNDRV\_PCM\_INFO\_MMAP |

SNDRV\_PCM\_INFO\_INTERLEAVED |

SNDRV\_PCM\_INFO\_BLOCK\_TRANSFER |

SNDRV\_PCM\_INFO\_MMAP\_VALID),

.formats = SNDRV\_PCM\_FMTBIT\_S16\_LE,

.rates = SNDRV\_PCM\_RATE\_8000\_48000,

.rate\_min = 8000,

.rate\_max = 48000,

.channels\_min = 2,

.channels\_max = 2,

.buffer\_bytes\_max = 32768,

.period\_bytes\_min = 4096,

.period\_bytes\_max = 32768,

.periods\_min = 1,

.periods\_max = 1024,

};

static struct snd\_pcm\_hardware snd\_mychip\_capture\_hw = {

.info = (SNDRV\_PCM\_INFO\_MMAP |

SNDRV\_PCM\_INFO\_INTERLEAVED |

SNDRV\_PCM\_INFO\_BLOCK\_TRANSFER |

SNDRV\_PCM\_INFO\_MMAP\_VALID),

.formats = SNDRV\_PCM\_FMTBIT\_S16\_LE,

.rates = SNDRV\_PCM\_RATE\_8000\_48000,

.rate\_min = 8000,

.rate\_max = 48000,

.channels\_min = 2,

.channels\_max = 2,

.buffer\_bytes\_max = 32768,

.period\_bytes\_min = 4096,

.period\_bytes\_max = 32768,

.periods\_min = 1,

.periods\_max = 1024,

};

1. 92c00941-51a5-46db-bdd4-fd681657228e
   1. 5x The function returns 1 when the fixup was successful, otherwise 0. The return value is used to update the statistics.
2. 0fef3653-0688-4cde-8644-8bce083eddcc
   1. 2x Avoid using this call until some usbcore bugs get fixed, since it does not fully synchronize device, interface, and driver (not just usbfs) state.
3. 345790bf-0e45-4448-8c12-0acd7301df9b
   1. Set a new mode, position and frame buffer. Depending on the device requirements, the mode can be stored internally by the driver and applied in the commit operation, or programmed to the hardware immediately. The mode\_set operation returns 0 on success or a negative error code if an error occurs. void (\*commit)(struct drm\_crtc \*crtc); Commit a mode. This operation is called after setting the new mode. Upon return the device must use the new mode and be fully operational.
   2. Set a new mode. Depending on the device requirements, the mode can be stored internally by the driver and applied in the commit operation, or programmed to the hardware immediately. void (\*commit)(struct drm\_encoder \*encoder); Commit a mode. This operation is called after setting the new mode. Upon return the device must use the new mode and be fully operational.
4. ddcae088-409a-482d-8e35-0e654d74fd0d
   1. You&#39;ll need to use spin\_lock() and spin\_unlock() for shared data.
   2. You&#39;ll need to use spin\_lock() and spin\_unlock() for shared data, whether it be a timer, tasklet, different softirq or the same or another softirq: any of them could be running on a different CPU.