

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-07/msg00449.html>

- *From:* Stefan Richter <stefanr@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 3 Jul 2006 01:02:21 +0200 (CEST)
-

Adjust tabulators, line wraps, empty lines, and comment style.
Update comments in ieee1394_transactions.h and highlevel.h.
Fix typo in comment in csr.h.

Signed-off-by: Stefan Richter <stefanr@xxxxxxxxxxxxxxxxxxxx>

```
---
drivers/ieee1394/csr.h | 96 +++-----
drivers/ieee1394/dma.h | 81 +++---
drivers/ieee1394/highlevel.h | 203 ++++++-----
drivers/ieee1394/hosts.h | 15 -
drivers/ieee1394/ieee1394-ioctl.h | 9
drivers/ieee1394/ieee1394.h | 312 ++++++-----
drivers/ieee1394/ieee1394_core.h | 17 -
drivers/ieee1394/ieee1394_transactions.h | 32 --
drivers/ieee1394/ieee1394_types.h | 31 +-
drivers/ieee1394/iso.h | 80 +++---
drivers/ieee1394/nodemgr.h | 16 -
11 files changed, 450 insertions(+), 442 deletions(-)
```

TODO:

Move comments from function declarations to function definitions.
Add kernel-doc compliant comments to data type definitions which are part of ieee1394's APIs like done in this patch for dma.h.

Index: linux/drivers/ieee1394/ieee1394_core.h

```
=====
--- linux.orig/drivers/ieee1394/ieee1394_core.h 2006-07-02 12:02:06.000000000 +0200
+++ linux/drivers/ieee1394/ieee1394_core.h 2006-07-02 12:16:52.000000000 +0200
@@ -58,7 +58,6 @@ struct hpsb_packet {
size_t header_size;
size_t data_size;

-
struct hpsb_host *host;
unsigned int generation;

@@ -80,7 +79,7 @@ struct hpsb_packet {
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
/* Set a task for when a packet completes */
void hpsb_set_packet_complete_task(struct hpsb_packet *packet,
- void (*routine)(void *), void *data);
+ void (*routine)(void *), void *data);

static inline struct hpsb_packet *driver_packet(struct list_head *)
{
@@ -92,7 +91,6 @@ void abort_timeouts(unsigned long __opa
struct hpsb_packet *hpsb_alloc_packet(size_t data_size);
void hpsb_free_packet(struct hpsb_packet *packet);

-
/*
* Generation counter for the complete 1394 subsystem. Generation gets
* incremented on every change in the subsystem (e.g. bus reset).
@@ -204,10 +202,14 @@ void hpsb_packet_received(struct hpsb_ho
#define IEEE1394_MINOR_BLOCK_EXPERIMENTAL 15

#define IEEE1394_CORE_DEV MKDEV(IEEE1394_MAJOR, 0)
-#define IEEE1394_RAW1394_DEV MKDEV(IEEE1394_MAJOR,
IEEE1394_MINOR_BLOCK_RAW1394 * 16)
-#define IEEE1394_VIDEO1394_DEV MKDEV(IEEE1394_MAJOR,
IEEE1394_MINOR_BLOCK_VIDEO1394 * 16)
-#define IEEE1394_DV1394_DEV MKDEV(IEEE1394_MAJOR, IEEE1394_MINOR_BLOCK_DV1394 *
16)
-#define IEEE1394_EXPERIMENTAL_DEV MKDEV(IEEE1394_MAJOR,
IEEE1394_MINOR_BLOCK_EXPERIMENTAL * 16)
+#define IEEE1394_RAW1394_DEV MKDEV(IEEE1394_MAJOR, \
+ IEEE1394_MINOR_BLOCK_RAW1394 * 16)
+#define IEEE1394_VIDEO1394_DEV MKDEV(IEEE1394_MAJOR, \
+ IEEE1394_MINOR_BLOCK_VIDEO1394 * 16)
+#define IEEE1394_DV1394_DEV MKDEV(IEEE1394_MAJOR, \
+ IEEE1394_MINOR_BLOCK_DV1394 * 16)
+#define IEEE1394_EXPERIMENTAL_DEV MKDEV(IEEE1394_MAJOR, \
+ IEEE1394_MINOR_BLOCK_EXPERIMENTAL * 16)

/* return the index (within a minor number block) of a file */
static inline unsigned char ieee1394_file_to_instance(struct file *file)
@@ -223,4 +225,3 @@ extern struct class hpsb_host_class;
extern struct class *hpsb_protocol_class;

#endif /* _IEEE1394_CORE_H */
-
Index: linux/drivers/ieee1394/ieee1394.h
=====
--- linux.orig/drivers/ieee1394/ieee1394.h 2006-07-02 12:11:23.000000000 +0200
+++ linux/drivers/ieee1394/ieee1394.h 2006-07-02 12:16:52.000000000 +0200
@@ -5,77 +5,76 @@
#ifndef _IEEE1394_IEEE1394_H
#define _IEEE1394_IEEE1394_H
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```

-#define TCODE_WRITEQ 0x0
-#define TCODE_WRITEB 0x1
-#define TCODE_WRITE_RESPONSE 0x2
-#define TCODE_READQ 0x4
-#define TCODE_READB 0x5
-#define TCODE_READQ_RESPONSE 0x6
-#define TCODE_READB_RESPONSE 0x7
-#define TCODE_CYCLE_START 0x8
-#define TCODE_LOCK_REQUEST 0x9
-#define TCODE_ISO_DATA 0xa
-#define TCODE_STREAM_DATA 0xa
-#define TCODE_LOCK_RESPONSE 0xb
-
-#define RCODE_COMPLETE 0x0
-#define RCODE_CONFLICT_ERROR 0x4
-#define RCODE_DATA_ERROR 0x5
-#define RCODE_TYPE_ERROR 0x6
-#define RCODE_ADDRESS_ERROR 0x7
-
-#define EXTCODE_MASK_SWAP 0x1
-#define EXTCODE_COMPARE_SWAP 0x2
-#define EXTCODE_FETCH_ADD 0x3
-#define EXTCODE_LITTLE_ADD 0x4
-#define EXTCODE_BOUNDED_ADD 0x5
-#define EXTCODE_WRAP_ADD 0x6
-
-#define ACK_COMPLETE 0x1
-#define ACK_PENDING 0x2
-#define ACK_BUSY_X 0x4
-#define ACK_BUSY_A 0x5
-#define ACK_BUSY_B 0x6
-#define ACK_TARDY 0xb
-#define ACK_CONFLICT_ERROR 0xc
-#define ACK_DATA_ERROR 0xd
-#define ACK_TYPE_ERROR 0xe
-#define ACK_ADDRESS_ERROR 0xf
+#define TCODE_WRITEQ 0x0
+#define TCODE_WRITEB 0x1
+#define TCODE_WRITE_RESPONSE 0x2
+#define TCODE_READQ 0x4
+#define TCODE_READB 0x5
+#define TCODE_READQ_RESPONSE 0x6
+#define TCODE_READB_RESPONSE 0x7
+#define TCODE_CYCLE_START 0x8
+#define TCODE_LOCK_REQUEST 0x9
+#define TCODE_ISO_DATA 0xa
+#define TCODE_STREAM_DATA 0xa
+#define TCODE_LOCK_RESPONSE 0xb
+
+#define RCODE_COMPLETE 0x0
+#define RCODE_CONFLICT_ERROR 0x4
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
+#define RCODE_DATA_ERROR 0x5
+#define RCODE_TYPE_ERROR 0x6
+#define RCODE_ADDRESS_ERROR 0x7
+
+#define EXTCODE_MASK_SWAP 0x1
+#define EXTCODE_COMPARE_SWAP 0x2
+#define EXTCODE_FETCH_ADD 0x3
+#define EXTCODE_LITTLE_ADD 0x4
+#define EXTCODE_BOUNDED_ADD 0x5
+#define EXTCODE_WRAP_ADD 0x6
+
+#define ACK_COMPLETE 0x1
+#define ACK_PENDING 0x2
+#define ACK_BUSY_X 0x4
+#define ACK_BUSY_A 0x5
+#define ACK_BUSY_B 0x6
+#define ACK_TARDY 0xb
+#define ACK_CONFLICT_ERROR 0xc
+#define ACK_DATA_ERROR 0xd
+#define ACK_TYPE_ERROR 0xe
+#define ACK_ADDRESS_ERROR 0xf

/* Non-standard "ACK codes" for internal use */
-#define ACKX_NONE (-1)
-#define ACKX_SEND_ERROR (-2)
-#define ACKX_ABORTED (-3)
-#define ACKX_TIMEOUT (-4)
+#define ACKX_NONE (-1)
+#define ACKX_SEND_ERROR (-2)
+#define ACKX_ABORTED (-3)
+#define ACKX_TIMEOUT (-4)
+
+#define IEEE1394_SPEED_100 0x00
+#define IEEE1394_SPEED_200 0x01
+#define IEEE1394_SPEED_400 0x02
+#define IEEE1394_SPEED_800 0x03
+#define IEEE1394_SPEED_1600 0x04
+#define IEEE1394_SPEED_3200 0x05

-
-#define IEEE1394_SPEED_100 0x00
-#define IEEE1394_SPEED_200 0x01
-#define IEEE1394_SPEED_400 0x02
-#define IEEE1394_SPEED_800 0x03
-#define IEEE1394_SPEED_1600 0x04
-#define IEEE1394_SPEED_3200 0x05
/* The current highest tested speed supported by the subsystem */
-#define IEEE1394_SPEED_MAX IEEE1394_SPEED_800
+#define IEEE1394_SPEED_MAX IEEE1394_SPEED_800

/* Maps speed values above to a string representation */
```

```
extern const char *hpsb_speedto_str[];

-
/* 1394a cable PHY packets */
-#define SELFID_PWRCL_NO_POWER 0x0
-#define SELFID_PWRCL_PROVIDE_15W 0x1
-#define SELFID_PWRCL_PROVIDE_30W 0x2
-#define SELFID_PWRCL_PROVIDE_45W 0x3
-#define SELFID_PWRCL_USE_1W 0x4
-#define SELFID_PWRCL_USE_3W 0x5
-#define SELFID_PWRCL_USE_6W 0x6
-#define SELFID_PWRCL_USE_10W 0x7
-
-#define SELFID_PORT_CHILD 0x3
-#define SELFID_PORT_PARENT 0x2
-#define SELFID_PORT_NCONN 0x1
-#define SELFID_PORT_NONE 0x0
+#define SELFID_PWRCL_NO_POWER 0x0
+#define SELFID_PWRCL_PROVIDE_15W 0x1
+#define SELFID_PWRCL_PROVIDE_30W 0x2
+#define SELFID_PWRCL_PROVIDE_45W 0x3
+#define SELFID_PWRCL_USE_1W 0x4
+#define SELFID_PWRCL_USE_3W 0x5
+#define SELFID_PWRCL_USE_6W 0x6
+#define SELFID_PWRCL_USE_10W 0x7
+
+#define SELFID_PORT_CHILD 0x3
+#define SELFID_PORT_PARENT 0x2
+#define SELFID_PORT_NCONN 0x1
+#define SELFID_PORT_NONE 0x0

#define SELFID_SPEED_UNKNOWN 0x3 /* 1394b PHY */

@@ -93,76 +92,76 @@ extern const char *hpsb_speedto_str[];

#define EXTPHYPACKET_TYPEMASK 0xC0FC0000

-#define PHYPACKET_PORT_SHIFT 24
-#define PHYPACKET_GAPCOUNT_SHIFT 16
+#define PHYPACKET_PORT_SHIFT 24
+#define PHYPACKET_GAPCOUNT_SHIFT 16

/* 1394a PHY register map bitmasks */
-#define PHY_00_PHYSICAL_ID 0xFC
-#define PHY_00_R 0x02 /* Root */
-#define PHY_00_PS 0x01 /* Power Status*/
-#define PHY_01_RHB 0x80 /* Root Hold-Off */
-#define PHY_01_IBR 0x80 /* Initiate Bus Reset */
-#define PHY_01_GAP_COUNT 0x3F
-#define PHY_02_EXTENDED 0xE0 /* 0x7 for 1394a-compliant PHY */
-#define PHY_02_TOTAL_PORTS 0x1F
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```

-#define PHY_03_MAX_SPEED 0xE0
-#define PHY_03_DELAY 0x0F
-#define PHY_04_LCTRL 0x80 /* Link Active Report Control */
-#define PHY_04_CONTENDER 0x40
-#define PHY_04_JITTER 0x38
-#define PHY_04_PWR_CLASS 0x07 /* Power Class */
-#define PHY_05_WATCHDOG 0x80
-#define PHY_05_ISBR 0x40 /* Initiate Short Bus Reset */
-#define PHY_05_LOOP 0x20 /* Loop Detect */
-#define PHY_05_PWR_FAIL 0x10 /* Cable Power Failure Detect */
-#define PHY_05_TIMEOUT 0x08 /* Arbitration State Machine Timeout */
-#define PHY_05_PORT_EVENT 0x04 /* Port Event Detect */
-#define PHY_05_ENAB_ACCEL 0x02 /* Enable Arbitration Acceleration */
-#define PHY_05_ENAB_MULTI 0x01 /* Ena. Multispeed Packet Concatenation */
+#define PHY_00_PHYSICAL_ID 0xFC
+#define PHY_00_R 0x02 /* Root */
+#define PHY_00_PS 0x01 /* Power Status*/
+#define PHY_01_RHB 0x80 /* Root Hold-Off */
+#define PHY_01_IBR 0x80 /* Initiate Bus Reset */
+#define PHY_01_GAP_COUNT 0x3F
+#define PHY_02_EXTENDED 0xE0 /* 0x7 for 1394a-compliant PHY */
+#define PHY_02_TOTAL_PORTS 0x1F
+#define PHY_03_MAX_SPEED 0xE0
+#define PHY_03_DELAY 0x0F
+#define PHY_04_LCTRL 0x80 /* Link Active Report Control */
+#define PHY_04_CONTENDER 0x40
+#define PHY_04_JITTER 0x38
+#define PHY_04_PWR_CLASS 0x07 /* Power Class */
+#define PHY_05_WATCHDOG 0x80
+#define PHY_05_ISBR 0x40 /* Initiate Short Bus Reset */
+#define PHY_05_LOOP 0x20 /* Loop Detect */
+#define PHY_05_PWR_FAIL 0x10 /* Cable Power Failure Detect */
+#define PHY_05_TIMEOUT 0x08 /* Arbitration State Machine Timeout */
+#define PHY_05_PORT_EVENT 0x04 /* Port Event Detect */
+#define PHY_05_ENAB_ACCEL 0x02 /* Enable Arbitration Acceleration */
+#define PHY_05_ENAB_MULTI 0x01 /* Ena. Multispeed Packet Concatenation */

#include <asm/byteorder.h>

#ifdef __BIG_ENDIAN_BITFIELD

struct selfid {
- u32 packet_identifier:2; /* always binary 10 */
- u32 phy_id:6;
- /* byte */
- u32 extended:1; /* if true is struct ext_selfid */
- u32 link_active:1;
- u32 gap_count:6;
- /* byte */
- u32 speed:2;
- u32 phy_delay:2;

```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
- u32 contender:1;
- u32 power_class:3;
- /* byte */
- u32 port0:2;
- u32 port1:2;
- u32 port2:2;
- u32 initiated_reset:1;
- u32 more_packets:1;
+ u32 packet_identifier:2; /* always binary 10 */
+ u32 phy_id:6;
+ /* byte */
+ u32 extended:1; /* if true is struct ext_selfid */
+ u32 link_active:1;
+ u32 gap_count:6;
+ /* byte */
+ u32 speed:2;
+ u32 phy_delay:2;
+ u32 contender:1;
+ u32 power_class:3;
+ /* byte */
+ u32 port0:2;
+ u32 port1:2;
+ u32 port2:2;
+ u32 initiated_reset:1;
+ u32 more_packets:1;
} __attribute__((packed));
```

```
struct ext_selfid {
- u32 packet_identifier:2; /* always binary 10 */
- u32 phy_id:6;
- /* byte */
- u32 extended:1; /* if false is struct selfid */
- u32 seq_nr:3;
- u32 reserved:2;
- u32 porta:2;
- /* byte */
- u32 portb:2;
- u32 portc:2;
- u32 portd:2;
- u32 porte:2;
- /* byte */
- u32 portf:2;
- u32 portg:2;
- u32 porth:2;
- u32 reserved2:1;
- u32 more_packets:1;
+ u32 packet_identifier:2; /* always binary 10 */
+ u32 phy_id:6;
+ /* byte */
+ u32 extended:1; /* if false is struct selfid */
+ u32 seq_nr:3;
```

```
+ u32 reserved:2;
+ u32 porta:2;
+ /* byte */
+ u32 portb:2;
+ u32 portc:2;
+ u32 portd:2;
+ u32 porte:2;
+ /* byte */
+ u32 portf:2;
+ u32 portg:2;
+ u32 porth:2;
+ u32 reserved2:1;
+ u32 more_packets:1;
} __attribute__((packed));

#ifdef __LITTLE_ENDIAN_BITFIELD /* __BIG_ENDIAN_BITFIELD */
@@ -173,49 +172,48 @@ struct ext_selfid {
*/

struct selfid {
- u32 phy_id:6;
- u32 packet_identifier:2; /* always binary 10 */
- /* byte */
- u32 gap_count:6;
- u32 link_active:1;
- u32 extended:1; /* if true is struct ext_selfid */
- /* byte */
- u32 power_class:3;
- u32 contender:1;
- u32 phy_delay:2;
- u32 speed:2;
- /* byte */
- u32 more_packets:1;
- u32 initiated_reset:1;
- u32 port2:2;
- u32 port1:2;
- u32 port0:2;
+ u32 phy_id:6;
+ u32 packet_identifier:2; /* always binary 10 */
+ /* byte */
+ u32 gap_count:6;
+ u32 link_active:1;
+ u32 extended:1; /* if true is struct ext_selfid */
+ /* byte */
+ u32 power_class:3;
+ u32 contender:1;
+ u32 phy_delay:2;
+ u32 speed:2;
+ /* byte */
+ u32 more_packets:1;
+ u32 initiated_reset:1;
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
+ u32 port2:2;
+ u32 port1:2;
+ u32 port0:2;
} __attribute__((packed));

struct ext_selfid {
- u32 phy_id:6;
- u32 packet_identifier:2; /* always binary 10 */
- /* byte */
- u32 porta:2;
- u32 reserved:2;
- u32 seq_nr:3;
- u32 extended:1; /* if false is struct selfid */
- /* byte */
- u32 porte:2;
- u32 portd:2;
- u32 portc:2;
- u32 portb:2;
- /* byte */
- u32 more_packets:1;
- u32 reserved2:1;
- u32 porth:2;
- u32 portg:2;
- u32 portf:2;
+ u32 phy_id:6;
+ u32 packet_identifier:2; /* always binary 10 */
+ /* byte */
+ u32 porta:2;
+ u32 reserved:2;
+ u32 seq_nr:3;
+ u32 extended:1; /* if false is struct selfid */
+ /* byte */
+ u32 porte:2;
+ u32 portd:2;
+ u32 portc:2;
+ u32 portb:2;
+ /* byte */
+ u32 more_packets:1;
+ u32 reserved2:1;
+ u32 porth:2;
+ u32 portg:2;
+ u32 portf:2;
} __attribute__((packed));

#else
#error What? PDP endian?
#endif /* __BIG_ENDIAN_BITFIELD */

-
#endif /* _IEEE1394_IEEE1394_H */
Index: linux/drivers/ieee1394/ieee1394-ioctl.h
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
=====
--- linux.orig/drivers/ieee1394/ieee1394-ioctl.h 2006-07-02 12:02:06.000000000 +0200
+++ linux/drivers/ieee1394/ieee1394-ioctl.h 2006-07-02 12:16:52.000000000 +0200
@@ -1,5 +1,7 @@
-/* Base file for all ieee1394 ioctl's. Linux-1394 has allocated base '#'
- * with a range of 0x00-0x3f. */
+/*
+ * Base file for all ieee1394 ioctl's.
+ * Linux-1394 has allocated base '#' with a range of 0x00-0x3f.
+ */

#ifdef __IEEE1394_IOCTL_H
#define __IEEE1394_IOCTL_H
@@ -96,8 +98,7 @@
_IOW ('#', 0x27, struct raw1394_iso_packets)
#define RAW1394_IOC_ISO_XMIT_SYNC \
_IO ('#', 0x28)
-#define RAW1394_IOC_ISO_RECV_FLUSH \
+#define RAW1394_IOC_ISO_RECV_FLUSH \
_IO ('#', 0x29)

-
#endif /* __IEEE1394_IOCTL_H */
Index: linux/drivers/ieee1394/ieee1394_transactions.h
=====
--- linux.orig/drivers/ieee1394/ieee1394_transactions.h 2006-07-02 12:02:06.000000000 +0200
+++ linux/drivers/ieee1394/ieee1394_transactions.h 2006-07-02 12:16:52.000000000 +0200
@@ -3,30 +3,25 @@

#include "ieee1394_core.h"

-
-/*
- * Get and free transaction labels.
- */
int hpsb_get_tlabel(struct hpsb_packet *packet);
void hpsb_free_tlabel(struct hpsb_packet *packet);

-
struct hpsb_packet *hpsb_make_readpacket(struct hpsb_host *host, nodeid_t node,
u64 addr, size_t length);
struct hpsb_packet *hpsb_make_lockpacket(struct hpsb_host *host, nodeid_t node,
- u64 addr, int extcode, quadlet_t *data,
+ u64 addr, int extcode, quadlet_t *data,
quadlet_t arg);
-struct hpsb_packet *hpsb_make_lock64packet(struct hpsb_host *host, nodeid_t node,
- u64 addr, int extcode, octlet_t *data,
- octlet_t arg);
-struct hpsb_packet *hpsb_make_phypacket(struct hpsb_host *host,
- quadlet_t data) ;
-struct hpsb_packet *hpsb_make_isopacket(struct hpsb_host *host,
- int length, int channel,
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```

- int tag, int sync);
- struct hpsb_packet *hpsb_make_writepacket (struct hpsb_host *host, nodeid_t node,
- u64 addr, quadlet_t *buffer, size_t length);
+ struct hpsb_packet *hpsb_make_lock64packet(struct hpsb_host *host,
+ nodeid_t node, u64 addr, int extcode,
+ octlet_t *data, octlet_t arg);
+ struct hpsb_packet *hpsb_make_phy packet(struct hpsb_host *host, quadlet_t data);
+ struct hpsb_packet *hpsb_make_isopacket(struct hpsb_host *host, int length,
+ int channel, int tag, int sync);
+ struct hpsb_packet *hpsb_make_writepacket(struct hpsb_host *host,
+ nodeid_t node, u64 addr,
+ quadlet_t *buffer, size_t length);
struct hpsb_packet *hpsb_make_streampacket(struct hpsb_host *host, u8 *buffer,
- int length, int channel, int tag, int sync);
+ int length, int channel, int tag,
+ int sync);

/*
 * hpsb_packet_success - Make sense of the ack and reply codes and
@@ -40,9 +35,8 @@ struct hpsb_packet *hpsb_make_streampack
*/
int hpsb_packet_success(struct hpsb_packet *packet);

-
/*
- * The generic read, write and lock functions. All recognize the local node ID
+ * The generic read and write functions. All recognize the local node ID
 * and act accordingly. Read and write automatically use quadlet commands if
 * length == 4 and and block commands otherwise (however, they do not yet
 * support lengths that are not a multiple of 4). You must explicitly specify
Index: linux/drivers/ieee1394/ieee1394_types.h
=====
--- linux.orig/drivers/ieee1394/ieee1394_types.h 2006-07-02 12:13:33.000000000 +0200
+++ linux/drivers/ieee1394/ieee1394_types.h 2006-07-02 12:16:52.000000000 +0200
@@ -31,7 +31,6 @@ do { \
sema_init(&(_tp)->count, 63); \
} while (0)

-
typedef u32 quadlet_t;
typedef u64 octlet_t;
typedef u16 nodeid_t;
@@ -54,16 +53,17 @@ typedef u16 arm_length_t;
#define NODE_BUS_ARGS(__host, __nodeid) \
__host->id, NODEID_TO_NODE(__nodeid), NODEID_TO_BUS(__nodeid)

-#define HPSB_PRINT(level, fmt, args...) printk(level "ieee1394: " fmt "\n" , ## args)
+#define HPSB_PRINT(level, fmt, args...) \
+ printk(level "ieee1394: " fmt "\n" , ## args)

-#define HPSB_DEBUG(fmt, args...) HPSB_PRINT(KERN_DEBUG, fmt , ## args)

```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```

-#define HPSB_INFO(fmt, args...) HPSB_PRINT(KERN_INFO, fmt , ## args)
-#define HPSB_NOTICE(fmt, args...) HPSB_PRINT(KERN_NOTICE, fmt , ## args)
-#define HPSB_WARN(fmt, args...) HPSB_PRINT(KERN_WARNING, fmt , ## args)
-#define HPSB_ERR(fmt, args...) HPSB_PRINT(KERN_ERR, fmt , ## args)
+#define HPSB_DEBUG(fmt, args...) HPSB_PRINT(KERN_DEBUG, fmt , ## args)
+#define HPSB_INFO(fmt, args...) HPSB_PRINT(KERN_INFO, fmt , ## args)
+#define HPSB_NOTICE(fmt, args...) HPSB_PRINT(KERN_NOTICE, fmt , ## args)
+#define HPSB_WARN(fmt, args...) HPSB_PRINT(KERN_WARNING, fmt , ## args)
+#define HPSB_ERR(fmt, args...) HPSB_PRINT(KERN_ERR, fmt , ## args)

```

```

#ifdef CONFIG_IEEE1394_VERBOSEDEBUG

```

```

-#define HPSB_VERBOSE(fmt, args...) HPSB_PRINT(KERN_DEBUG, fmt , ## args)
+#define HPSB_VERBOSE(fmt, args...) HPSB_PRINT(KERN_DEBUG, fmt , ## args)

```

```

#else

```

```

#define HPSB_VERBOSE(fmt, args...)

```

```

#endif

```

```

@@ -77,23 +77,20 @@ typedef u16 arm_length_t;

```

```

static inline void *memcpy_le32(u32 *dest, const u32 *__src, size_t count)

```

```

{
- void *tmp = dest;
+ void *tmp = dest;
u32 *src = (u32 *)__src;

- count /= 4;
-
- while (count--) {
- *dest++ = swab32p(src++);
- }
-
- return tmp;
+ count /= 4;
+ while (count--)
+ *dest++ = swab32p(src++);
+ return tmp;
}

```

```

#else

```

```

static __inline__ void *memcpy_le32(u32 *dest, const u32 *src, size_t count)

```

```

{
- return memcpy(dest, src, count);
+ return memcpy(dest, src, count);
}

```

```

#endif /* __BIG_ENDIAN */

```

```

Index: linux/drivers/ieee1394/csr.h

```

```

=====

```

```

--- linux.orig/drivers/ieee1394/csr.h 2006-07-02 12:02:06.000000000 +0200

```

```

+++ linux/drivers/ieee1394/csr.h 2006-07-02 12:16:52.000000000 +0200

```

```

@@ -8,68 +8,68 @@

```

```
#include "csr1212.h"

-#define CSR_REGISTER_BASE 0xfffff0000000ULL
+#define CSR_REGISTER_BASE 0xfffff0000000ULL

/* register offsets relative to CSR_REGISTER_BASE */
-#define CSR_STATE_CLEAR 0x0
-#define CSR_STATE_SET 0x4
-#define CSR_NODE_IDS 0x8
-#define CSR_RESET_START 0xc
-#define CSR_SPLIT_TIMEOUT_HI 0x18
-#define CSR_SPLIT_TIMEOUT_LO 0x1c
-#define CSR_CYCLE_TIME 0x200
-#define CSR_BUS_TIME 0x204
-#define CSR_BUSY_TIMEOUT 0x210
-#define CSR_BUS_MANAGER_ID 0x21c
-#define CSR_BANDWIDTH_AVAILABLE 0x220
-#define CSR_CHANNELS_AVAILABLE 0x224
-#define CSR_CHANNELS_AVAILABLE_HI 0x224
-#define CSR_CHANNELS_AVAILABLE_LO 0x228
-#define CSR_BROADCAST_CHANNEL 0x234
-#define CSR_CONFIG_ROM 0x400
-#define CSR_CONFIG_ROM_END 0x800
-#define CSR_FCP_COMMAND 0xB00
-#define CSR_FCP_RESPONSE 0xD00
-#define CSR_FCP_END 0xF00
-#define CSR_TOPOLOGY_MAP 0x1000
-#define CSR_TOPOLOGY_MAP_END 0x1400
-#define CSR_SPEED_MAP 0x2000
-#define CSR_SPEED_MAP_END 0x3000
+#define CSR_STATE_CLEAR 0x0
+#define CSR_STATE_SET 0x4
+#define CSR_NODE_IDS 0x8
+#define CSR_RESET_START 0xc
+#define CSR_SPLIT_TIMEOUT_HI 0x18
+#define CSR_SPLIT_TIMEOUT_LO 0x1c
+#define CSR_CYCLE_TIME 0x200
+#define CSR_BUS_TIME 0x204
+#define CSR_BUSY_TIMEOUT 0x210
+#define CSR_BUS_MANAGER_ID 0x21c
+#define CSR_BANDWIDTH_AVAILABLE 0x220
+#define CSR_CHANNELS_AVAILABLE 0x224
+#define CSR_CHANNELS_AVAILABLE_HI 0x224
+#define CSR_CHANNELS_AVAILABLE_LO 0x228
+#define CSR_BROADCAST_CHANNEL 0x234
+#define CSR_CONFIG_ROM 0x400
+#define CSR_CONFIG_ROM_END 0x800
+#define CSR_FCP_COMMAND 0xB00
+#define CSR_FCP_RESPONSE 0xD00
+#define CSR_FCP_END 0xF00
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
+#define CSR_TOPOLOGY_MAP 0x1000
+#define CSR_TOPOLOGY_MAP_END 0x1400
+#define CSR_SPEED_MAP 0x2000
+#define CSR_SPEED_MAP_END 0x3000

/* IEEE 1394 bus specific Configuration ROM Key IDs */
#define IEEE1394_KV_ID_POWER_REQUIREMENTS (0x30)

-/* IEEE 1394 Bus Information Block specifics */
+/* IEEE 1394 Bus Information Block specifics */
#define CSR_BUS_INFO_SIZE (5 * sizeof(quadlet_t))

-#define CSR_IRMC_SHIFT 31
-#define CSR_CMC_SHIFT 30
-#define CSR_ISC_SHIFT 29
-#define CSR_BMC_SHIFT 28
-#define CSR_PMC_SHIFT 27
-#define CSR_CYC_CLK_ACC_SHIFT 16
-#define CSR_MAX_REC_SHIFT 12
-#define CSR_MAX_ROM_SHIFT 8
-#define CSR_GENERATION_SHIFT 4
+#define CSR_IRMC_SHIFT 31
+#define CSR_CMC_SHIFT 30
+#define CSR_ISC_SHIFT 29
+#define CSR_BMC_SHIFT 28
+#define CSR_PMC_SHIFT 27
+#define CSR_CYC_CLK_ACC_SHIFT 16
+#define CSR_MAX_REC_SHIFT 12
+#define CSR_MAX_ROM_SHIFT 8
+#define CSR_GENERATION_SHIFT 4

#define CSR_SET_BUS_INFO_GENERATION(csr, gen) \
((csr)->bus_info_data[2] = \
cpu_to_be32((be32_to_cpu((csr)->bus_info_data[2]) & \
~(0xf << CSR_GENERATION_SHIFT)) | \
+(gen) << CSR_GENERATION_SHIFT))

struct csr_control {
- spinlock_t lock;
+ spinlock_t lock;

- quadlet_t state;
- quadlet_t node_ids;
- quadlet_t split_timeout_hi, split_timeout_lo;
- unsigned long expire; // Calculated from split_timeout
- quadlet_t cycle_time;
- quadlet_t bus_time;
- quadlet_t bus_manager_id;
- quadlet_t bandwidth_available;
- quadlet_t channels_available_hi, channels_available_lo;
```

```
+ quadlet_t state;
+ quadlet_t node_ids;
+ quadlet_t split_timeout_hi, split_timeout_lo;
+ unsigned long expire; /* Calculated from split_timeout */
+ quadlet_t cycle_time;
+ quadlet_t bus_time;
+ quadlet_t bus_manager_id;
+ quadlet_t bandwidth_available;
+ quadlet_t channels_available_hi, channels_available_lo;
quadlet_t broadcast_channel;
```

```
/* Bus Info */
```

```
@@ -84,8 +84,8 @@ struct csr_control {
```

```
struct csr1212_csr *rom;
```

```
- quadlet_t topology_map[256];
- quadlet_t speed_map[1024];
+ quadlet_t topology_map[256];
+ quadlet_t speed_map[1024];
};
```

```
extern struct csr1212_bus_ops csr_bus_ops;
```

```
Index: linux/drivers/ieee1394/dma.h
```

```
----- linux.orig/drivers/ieee1394/dma.h 2006-07-02 12:02:06.000000000 +0200
```

```
+++ linux/drivers/ieee1394/dma.h 2006-07-02 12:16:52.000000000 +0200
```

```
@@ -13,66 +13,85 @@
```

```
#include <linux/pci.h>
```

```
#include <asm/scatterlist.h>
```

```
-/* struct dma_prog_region
```

```
-
```

```
- a small, physically-contiguous DMA buffer with random-access,
```

```
- synchronous usage characteristics
```

```
*/
```

```
-
```

```
+/**
```

```
+ * struct dma_prog_region - small contiguous DMA buffer
```

```
+ * @kvirt: kernel virtual address
```

```
+ * @dev: PCI device
```

```
+ * @n_pages: number of kernel pages
```

```
+ * @bus_addr: base bus address
```

```
+ *
```

```
+ * a small, physically contiguous DMA buffer with random-access, synchronous
```

```
+ * usage characteristics
```

```
+ */
```

```
struct dma_prog_region {
```

```
- unsigned char *kvirt; /* kernel virtual address */
```

```
- struct pci_dev *dev; /* PCI device */
```

```
- unsigned int n_pages; /* # of kernel pages */
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
– dma_addr_t bus_addr; /* base bus address */
+ unsigned char *kvirt;
+ struct pci_dev *dev;
+ unsigned int n_pages;
+ dma_addr_t bus_addr;
};

/* clear out all fields but do not allocate any memory */
void dma_prog_region_init(struct dma_prog_region *prog);
–int dma_prog_region_alloc(struct dma_prog_region *prog, unsigned long n_bytes, struct pci_dev *dev);
+int dma_prog_region_alloc(struct dma_prog_region *prog, unsigned long n_bytes,
+ struct pci_dev *dev);
void dma_prog_region_free(struct dma_prog_region *prog);

–static inline dma_addr_t dma_prog_region_offset_to_bus(struct dma_prog_region *prog, unsigned long
offset)
+static inline dma_addr_t dma_prog_region_offset_to_bus(
+ struct dma_prog_region *prog, unsigned long offset)
{
return prog->bus_addr + offset;
}

–/* struct dma_region
–
– a large, non-physically-contiguous DMA buffer with streaming,
– asynchronous usage characteristics
–*/
–
+/**
+ * struct dma_region – large non-contiguous DMA buffer
+ * @virt: kernel virtual address
+ * @dev: PCI device
+ * @n_pages: number of kernel pages
+ * @n_dma_pages: number of IOMMU pages
+ * @sglist: IOMMU mapping
+ * @direction: PCI_DMA_TODEVICE, etc.
+ *
+ * a large, non-physically-contiguous DMA buffer with streaming, asynchronous
+ * usage characteristics
+ */
struct dma_region {
– unsigned char *kvirt; /* kernel virtual address */
– struct pci_dev *dev; /* PCI device */
– unsigned int n_pages; /* # of kernel pages */
– unsigned int n_dma_pages; /* # of IOMMU pages */
– struct scatterlist *sglist; /* IOMMU mapping */
– int direction; /* PCI_DMA_TODEVICE, etc */
+ unsigned char *kvirt;
+ struct pci_dev *dev;
+ unsigned int n_pages;
+ unsigned int n_dma_pages;
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
+ struct scatterlist *sglist;
+ int direction;
};

/* clear out all fields but do not allocate anything */
void dma_region_init(struct dma_region *dma);

/* allocate the buffer and map it to the IOMMU */
-int dma_region_alloc(struct dma_region *dma, unsigned long n_bytes, struct pci_dev *dev, int direction);
+int dma_region_alloc(struct dma_region *dma, unsigned long n_bytes,
+ struct pci_dev *dev, int direction);

/* unmap and free the buffer */
void dma_region_free(struct dma_region *dma);

/* sync the CPU's view of the buffer */
-void dma_region_sync_for_cpu(struct dma_region *dma, unsigned long offset, unsigned long len);
+void dma_region_sync_for_cpu(struct dma_region *dma, unsigned long offset,
+ unsigned long len);
+
/* sync the IO bus' view of the buffer */
-void dma_region_sync_for_device(struct dma_region *dma, unsigned long offset, unsigned long len);
+void dma_region_sync_for_device(struct dma_region *dma, unsigned long offset,
+ unsigned long len);

/* map the buffer into a user space process */
-int dma_region_mmap(struct dma_region *dma, struct file *file, struct vm_area_struct *vma);
+int dma_region_mmap(struct dma_region *dma, struct file *file,
+ struct vm_area_struct *vma);

/* macro to index into a DMA region (or dma_prog_region) */
-#define dma_region_i(_dma, _type, _index) ( ((_type*) ((_dma)->kvirt)) + (_index) )
+#define dma_region_i(_dma, _type, _index) \
+ ( ((_type*) ((_dma)->kvirt)) + (_index) )

/* return the DMA bus address of the byte with the given offset
- relative to the beginning of the dma_region */
-dma_addr_t dma_region_offset_to_bus(struct dma_region *dma, unsigned long offset);
+ * relative to the beginning of the dma_region */
+dma_addr_t dma_region_offset_to_bus(struct dma_region *dma,
+ unsigned long offset);

#endif /* IEEE1394_DMA_H */
Index: linux/drivers/ieee1394/highlevel.h
=====
--- linux.orig/drivers/ieee1394/highlevel.h 2006-07-02 12:02:06.000000000 +0200
+++ linux/drivers/ieee1394/highlevel.h 2006-07-02 12:16:52.000000000 +0200
@@ -1,60 +1,51 @@
-
-#ifndef IEEE1394_HIGHLEVEL_H
-#define IEEE1394_HIGHLEVEL_H
```

```

-
+/* internal to ieee1394 core */
struct hpsb_address_serve {
- struct list_head host_list; /* per host list */
-
- struct list_head hl_list; /* hpsb_highlevel list */
-
- struct hpsb_address_ops *op;
-
+ struct list_head host_list; /* per host list */
+ struct list_head hl_list; /* hpsb_highlevel list */
+ struct hpsb_address_ops *op;
struct hpsb_host *host;
-
- /* first address handled and first address behind, quadlet aligned */
- u64 start, end;
+ u64 start; /* first address handled, quadlet aligned */
+ u64 end; /* first address behind, quadlet aligned */
};

-
-/*
- * The above structs are internal to highlevel driver handling. Only the
- * following structures are of interest to actual highlevel drivers.
- */
+/* Only the following structures are of interest to actual highlevel drivers. */

struct hpsb_highlevel {
struct module *owner;
const char *name;

- /* Any of the following pointers can legally be NULL, except for
- * iso_receive which can only be NULL when you don't request
- * channels. */
-
- /* New host initialized. Will also be called during
- * hpsb_register_highlevel for all hosts already installed. */
- void (*add_host) (struct hpsb_host *host);
-
- /* Host about to be removed. Will also be called during
- * hpsb_unregister_highlevel once for each host. */
- void (*remove_host) (struct hpsb_host *host);
+ /* Any of the following pointers can legally be NULL, except for
+ * iso_receive which can only be NULL when you don't request
+ * channels. */
+
+ /* New host initialized. Will also be called during
+ * hpsb_register_highlevel for all hosts already installed. */
+ void (*add_host)(struct hpsb_host *host);
+
+

```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
+ /* Host about to be removed. Will also be called during
+ * hpsb_unregister_highlevel once for each host. */
+ void (*remove_host)(struct hpsb_host *host);

- /* Host experienced bus reset with possible configuration changes.
+ /* Host experienced bus reset with possible configuration changes.
* Note that this one may occur during interrupt/bottom half handling.
* You can not expect to be able to do stock hpsb_reads. */
- void (*host_reset) (struct hpsb_host *host);
+ void (*host_reset)(struct hpsb_host *host);

- /* An isochronous packet was received. Channel contains the channel
- * number for your convenience, it is also contained in the included
- * packet header (first quadlet, CRCs are missing). You may get called
- * for channel/host combinations you did not request. */
- void (*iso_receive) (struct hpsb_host *host, int channel,
- quadlet_t *data, size_t length);
-
- /* A write request was received on either the FCP_COMMAND (direction =
- * 0) or the FCP_RESPONSE (direction = 1) register. The cts arg
- * contains the cts field (first byte of data). */
- void (*fcp_request) (struct hpsb_host *host, int nodeid, int direction,
- int cts, u8 *data, size_t length);
+ /* An isochronous packet was received. Channel contains the channel
+ * number for your convenience, it is also contained in the included
+ * packet header (first quadlet, CRCs are missing). You may get called
+ * for channel/host combinations you did not request. */
+ void (*iso_receive)(struct hpsb_host *host, int channel,
+ quadlet_t *data, size_t length);
+
+ /* A write request was received on either the FCP_COMMAND (direction =
+ * 0) or the FCP_RESPONSE (direction = 1) register. The cts arg
+ * contains the cts field (first byte of data). */
+ void (*fcp_request)(struct hpsb_host *host, int nodeid, int direction,
+ int cts, u8 *data, size_t length);

/* These are initialized by the subsystem when the
* hpsb_highlevel is registered. */
@@ -67,61 +58,62 @@ struct hpsb_highlevel {
};

struct hpsb_address_ops {
- /*
- * Null function pointers will make the respective operation complete
- * with RCODE_TYPE_ERROR. Makes for easy to implement read-only
- * registers (just leave everything but read NULL).
- *
- * All functions shall return appropriate IEEE 1394 rcodes.
- */
-
- /* These functions have to implement block reads for themselves. */
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
- /* These functions either return a response code
- or a negative number. In the first case a response will be generated; in the
- later case, no response will be sent and the driver, that handled the request
- will send the response itself
- */
- int (*read) (struct hpsb_host *host, int nodeid, quadlet_t *buffer,
- u64 addr, size_t length, u16 flags);
- int (*write) (struct hpsb_host *host, int nodeid, int destid,
- quadlet_t *data, u64 addr, size_t length, u16 flags);
-
- /* Lock transactions: write results of ext_tcode operation into
- * *store. */
- int (*lock) (struct hpsb_host *host, int nodeid, quadlet_t *store,
- u64 addr, quadlet_t data, quadlet_t arg, int ext_tcode, u16 flags);
- int (*lock64) (struct hpsb_host *host, int nodeid, octlet_t *store,
- u64 addr, octlet_t data, octlet_t arg, int ext_tcode, u16 flags);
+ /*
+ * Null function pointers will make the respective operation complete
+ * with RCODE_TYPE_ERROR. Makes for easy to implement read-only
+ * registers (just leave everything but read NULL).
+ *
+ * All functions shall return appropriate IEEE 1394 rcodes.
+ */
+
+ /* These functions have to implement block reads for themselves.
+ *
+ * These functions either return a response code or a negative number.
+ * In the first case a response will be generated. In the latter case,
+ * no response will be sent and the driver which handled the request
+ * will send the response itself. */
+ int (*read)(struct hpsb_host *host, int nodeid, quadlet_t *buffer,
+ u64 addr, size_t length, u16 flags);
+ int (*write)(struct hpsb_host *host, int nodeid, int destid,
+ quadlet_t *data, u64 addr, size_t length, u16 flags);
+
+ /* Lock transactions: write results of ext_tcode operation into
+ * *store. */
+ int (*lock)(struct hpsb_host *host, int nodeid, quadlet_t *store,
+ u64 addr, quadlet_t data, quadlet_t arg, int ext_tcode,
+ u16 flags);
+ int (*lock64)(struct hpsb_host *host, int nodeid, octlet_t *store,
+ u64 addr, octlet_t data, octlet_t arg, int ext_tcode,
+ u16 flags);
+ };
-
- void highlevel_add_host(struct hpsb_host *host);
- void highlevel_remove_host(struct hpsb_host *host);
- void highlevel_host_reset(struct hpsb_host *host);
-
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```

-/* these functions are called to handle transactions. They are called, when
- a packet arrives. The flags argument contains the second word of the first header
- quadlet of the incoming packet (containing transaction label, retry code,
- transaction code and priority). These functions either return a response code
- or a negative number. In the first case a response will be generated; in the
- later case, no response will be sent and the driver, that handled the request
- will send the response itself.
-*/
-int highlevel_read(struct hpsb_host *host, int nodeid, void *data,
- u64 addr, unsigned int length, u16 flags);
-int highlevel_write(struct hpsb_host *host, int nodeid, int destid,
- void *data, u64 addr, unsigned int length, u16 flags);
+/*
+ * These functions are called to handle transactions. They are called when a
+ * packet arrives. The flags argument contains the second word of the first
+ * header quadlet of the incoming packet (containing transaction label, retry
+ * code, transaction code and priority). These functions either return a
+ * response code or a negative number. In the first case a response will be
+ * generated. In the latter case, no response will be sent and the driver which
+ * handled the request will send the response itself.
+ */
+int highlevel_read(struct hpsb_host *host, int nodeid, void *data, u64 addr,
+ unsigned int length, u16 flags);
+int highlevel_write(struct hpsb_host *host, int nodeid, int destid, void *data,
+ u64 addr, unsigned int length, u16 flags);
int highlevel_lock(struct hpsb_host *host, int nodeid, quadlet_t *store,
- u64 addr, quadlet_t data, quadlet_t arg, int ext_tcode, u16 flags);
+ u64 addr, quadlet_t data, quadlet_t arg, int ext_tcode,
+ u16 flags);
int highlevel_lock64(struct hpsb_host *host, int nodeid, octlet_t *store,
- u64 addr, octlet_t data, octlet_t arg, int ext_tcode, u16 flags);
+ u64 addr, octlet_t data, octlet_t arg, int ext_tcode,
+ u16 flags);

-void highlevel_iso_receive(struct hpsb_host *host, void *data,
- size_t length);
+void highlevel_iso_receive(struct hpsb_host *host, void *data, size_t length);
void highlevel_fcp_request(struct hpsb_host *host, int nodeid, int direction,
- void *data, size_t length);
-
+ void *data, size_t length);

/*
* Register highlevel driver. The name pointer has to stay valid at all times
@@ -132,13 +124,15 @@ void hpsb_unregister_highlevel(struct hp

/*
* Register handlers for host address spaces. Start and end are 48 bit pointers
- * and have to be quadlet aligned (end points to the first address behind the
- * handled addresses. This function can be called multiple times for a single
- * hpsb_highlevel to implement sparse register sets. The requested region must

```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
- * not overlap any previously allocated region, otherwise registering will fail.
+ * and have to be quadlet aligned. Argument "end" points to the first address
+ * behind the handled addresses. This function can be called multiple times for
+ * a single hpsb_highlevel to implement sparse register sets. The requested
+ * region must not overlap any previously allocated region, otherwise
+ * registering will fail.
*
- * It returns true for successful allocation. There is no unregister function,
- * all address spaces are deallocated together with the hpsb_highlevel.
+ * It returns true for successful allocation. Address spaces can be
+ * unregistered with hpsb_unregister_addrspace. All remaining address spaces
+ * are automatically deallocated together with the hpsb_highlevel.
*/
u64 hpsb_allocate_and_register_addrspace(struct hpsb_highlevel *hl,
struct hpsb_host *host,
@@ -146,20 +140,18 @@ u64 hpsb_allocate_and_register_addrspace
u64 size, u64 alignment,
u64 start, u64 end);
int hpsb_register_addrspace(struct hpsb_highlevel *hl, struct hpsb_host *host,
- struct hpsb_address_ops *ops, u64 start, u64 end);
-
+ struct hpsb_address_ops *ops, u64 start, u64 end);
int hpsb_unregister_addrspace(struct hpsb_highlevel *hl, struct hpsb_host *host,
- u64 start);
+ u64 start);

/*
* Enable or disable receiving a certain isochronous channel through the
* iso_receive op.
*/
int hpsb_listen_channel(struct hpsb_highlevel *hl, struct hpsb_host *host,
- unsigned int channel);
+ unsigned int channel);
void hpsb_unlisten_channel(struct hpsb_highlevel *hl, struct hpsb_host *host,
- unsigned int channel);
-
+ unsigned int channel);

/* Retrieve a hostinfo pointer bound to this driver/host */
void *hpsb_get_hostinfo(struct hpsb_highlevel *hl, struct hpsb_host *host);
@@ -172,19 +164,24 @@ void *hpsb_create_hostinfo(struct hpsb_h
void hpsb_destroy_hostinfo(struct hpsb_highlevel *hl, struct hpsb_host *host);

/* Set an alternate lookup key for the hostinfo bound to this driver/host */
-void hpsb_set_hostinfo_key(struct hpsb_highlevel *hl, struct hpsb_host *host, unsigned long key);
+void hpsb_set_hostinfo_key(struct hpsb_highlevel *hl, struct hpsb_host *host,
+ unsigned long key);

-/* Retrieve the alternate lookup key for the hostinfo bound to this driver/host */
-unsigned long hpsb_get_hostinfo_key(struct hpsb_highlevel *hl, struct hpsb_host *host);
+/* Retrieve the alternate lookup key for the hostinfo bound to this
```

```

+ * driver/host */
+unsigned long hpsb_get_hostinfo_key(struct hpsb_highlevel *hl,
+ struct hpsb_host *host);

/* Retrieve a hostinfo pointer bound to this driver using its alternate key */
void *hpsb_get_hostinfo_bykey(struct hpsb_highlevel *hl, unsigned long key);

/* Set the hostinfo pointer to something useful. Usually follows a call to
* hpsb_create_hostinfo, where the size is 0. */
-int hpsb_set_hostinfo(struct hpsb_highlevel *hl, struct hpsb_host *host, void *data);
+int hpsb_set_hostinfo(struct hpsb_highlevel *hl, struct hpsb_host *host,
+ void *data);

/* Retrieve hpsb_host using a highlevel handle and a key */
-struct hpsb_host *hpsb_get_host_bykey(struct hpsb_highlevel *hl, unsigned long key);
+struct hpsb_host *hpsb_get_host_bykey(struct hpsb_highlevel *hl,
+ unsigned long key);

#endif /* IEEE1394_HIGHLEVEL_H */
Index: linux/drivers/ieee1394/hosts.h
=====
--- linux.orig/drivers/ieee1394/hosts.h 2006-07-02 12:02:06.000000000 +0200
+++ linux/drivers/ieee1394/hosts.h 2006-07-02 12:16:52.000000000 +0200
@@ -112,7 +112,7 @@ enum devctl_cmd {

enum isoctl_cmd {
/* rawiso API - see iso.h for the meanings of these commands
- (they correspond exactly to the hpsb_iso_* API functions)
+ * (they correspond exactly to the hpsb_iso_* API functions)
* INIT = allocate resources
* START = begin transmission/reception
* STOP = halt transmission/reception
@@ -160,7 +160,8 @@ struct hpsb_host_driver {
/* The hardware driver may optionally support a function that is used
* to set the hardware ConfigROM if the hardware supports handling
* reads to the ConfigROM on its own. */
- void (*set_hw_config_rom) (struct hpsb_host *host, quadlet_t *config_rom);
+ void (*set_hw_config_rom)(struct hpsb_host *host,
+ quadlet_t *config_rom);

/* This function shall implement packet transmission based on
* packet->type. It shall CRC both parts of the packet (unless
@@ -170,20 +171,21 @@ struct hpsb_host_driver {
* called. Return 0 on success, negative errno on failure.
* NOTE: The function must be callable in interrupt context.
*/
- int (*transmit_packet) (struct hpsb_host *host,
- struct hpsb_packet *packet);
+ int (*transmit_packet)(struct hpsb_host *host,
+ struct hpsb_packet *packet);

```

```

/* This function requests miscellaneous services from the driver, see
 * above for command codes and expected actions. Return -1 for unknown
 * command, though that should never happen.
 */
- int (*devctl) (struct hpsb_host *host, enum devctl_cmd command, int arg);
+ int (*devctl)(struct hpsb_host *host, enum devctl_cmd command, int arg);

/* ISO transmission/reception functions. Return 0 on success, -1
 * (or -EXXX errno code) on failure. If the low-level driver does not
 * support the new ISO API, set isoctl to NULL.
 */
- int (*isoctl) (struct hpsb_iso *iso, enum isoctl_cmd command, unsigned long arg);
+ int (*isoctl)(struct hpsb_iso *iso, enum isoctl_cmd command,
+ unsigned long arg);

/* This function is mainly to redirect local CSR reads/locks to the iso
 * management registers (bus manager id, bandwidth available, channels
 * @@ -196,7 +198,6 @@ struct hpsb_host_driver {
quadlet_t data, quadlet_t compare);
};

-
struct hpsb_host *hpsb_alloc_host(struct hpsb_host_driver *drv, size_t extra,
struct device *dev);
int hpsb_add_host(struct hpsb_host *host);
Index: linux/drivers/ieee1394/iso.h
=====
--- linux.orig/drivers/ieee1394/iso.h 2006-07-02 12:02:06.000000000 +0200
+++ linux/drivers/ieee1394/iso.h 2006-07-02 12:16:52.000000000 +0200
@@ -17,28 +17,30 @@

/* high-level ISO interface */

-/* This API sends and receives isochronous packets on a large,
- virtually-contiguous kernel memory buffer. The buffer may be mapped
- into a user-space process for zero-copy transmission and reception.
-
- There are no explicit boundaries between packets in the buffer. A
- packet may be transmitted or received at any location. However,
- low-level drivers may impose certain restrictions on alignment or
- size of packets. (e.g. in OHCI no packet may cross a page boundary,
- and packets should be quadlet-aligned)
-*/
+/*
+ * This API sends and receives isochronous packets on a large,
+ * virtually-contiguous kernel memory buffer. The buffer may be mapped
+ * into a user-space process for zero-copy transmission and reception.
+ *
+ * There are no explicit boundaries between packets in the buffer. A
+ * packet may be transmitted or received at any location. However,
+ * low-level drivers may impose certain restrictions on alignment or

```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
+ * size of packets. (e.g. in OHCI no packet may cross a page boundary,  
+ * and packets should be quadlet-aligned)  
+ */  
  
/* Packet descriptor – the API maintains a ring buffer of these packet  
– descriptors in kernel memory (hpsb_iso.infos[]). */  
–  
+ * descriptors in kernel memory (hpsb_iso.infos[]). */  
struct hpsb_iso_packet_info {  
/* offset of data payload relative to the first byte of the buffer */  
__u32 offset;  
  
– /* length of the data payload, in bytes (not including the isochronous header) */  
+ /* length of the data payload, in bytes (not including the isochronous  
+ * header) */  
__u16 len;  
  
– /* (recv only) the cycle number (mod 8000) on which the packet was received */  
+ /* (recv only) the cycle number (mod 8000) on which the packet was  
+ * received */  
__u16 cycle;  
  
/* (recv only) channel on which the packet was received */  
@@ –48,12 +50,10 @@ struct hpsb_iso_packet_info {  
__u8 tag;  
__u8 sy;  
  
– /*  
– * length in bytes of the packet including header/trailer.  
– * MUST be at structure end, since the first part of this structure is also  
– * defined in raw1394.h (i.e. struct raw1394_iso_packet_info), is copied to  
– * userspace and is accessed there through libraw1394.  
– */  
+ /* length in bytes of the packet including header/trailer.  
+ * MUST be at structure end, since the first part of this structure is  
+ * also defined in raw1394.h (i.e. struct raw1394_iso_packet_info), is  
+ * copied to userspace and is accessed there through libraw1394. */  
__u16 total_len;  
};  
  
@@ –75,8 +75,8 @@ struct hpsb_iso {  
void *hostdata;  
  
/* a function to be called (from interrupt context) after  
– outgoing packets have been sent, or incoming packets have  
– arrived */  
+ * outgoing packets have been sent, or incoming packets have  
+ * arrived */  
void (*callback)(struct hpsb_iso*);  
  
/* wait for buffer space */
```

```

@@ -88,7 +88,7 @@ struct hpsb_iso {

/* greatest # of packets between interrupts – controls
– the maximum latency of the buffer */
+ * the maximum latency of the buffer */
int irq_interval;

/* the buffer for packet data payloads */
@@ -112,8 +112,8 @@ struct hpsb_iso {
int pkt_dma;

/* how many packets, starting at first_packet:
– (transmit) are ready to be filled with data
– (receive) contain received data */
+ * (transmit) are ready to be filled with data
+ * (receive) contain received data */
int n_ready_packets;

/* how many times the buffer has overflowed or underflowed */
@@ -134,7 +134,7 @@ struct hpsb_iso {
int start_cycle;

/* cycle at which next packet will be transmitted,
– -1 if not known */
+ * -1 if not known */
int xmit_cycle;

/* ringbuffer of packet descriptors in regular kernel memory
@@ -170,25 +170,30 @@ int hpsb_iso_recv_unlisten_channel(struct
int hpsb_iso_recv_set_channel_mask(struct hpsb_iso *iso, u64 mask);

/* start/stop DMA */
-int hpsb_iso_xmit_start(struct hpsb_iso *iso, int start_on_cycle, int prebuffer);
-int hpsb_iso_recv_start(struct hpsb_iso *iso, int start_on_cycle, int tag_mask, int sync);
+int hpsb_iso_xmit_start(struct hpsb_iso *iso, int start_on_cycle,
+ int prebuffer);
+int hpsb_iso_recv_start(struct hpsb_iso *iso, int start_on_cycle,
+ int tag_mask, int sync);
void hpsb_iso_stop(struct hpsb_iso *iso);

/* deallocate buffer and DMA context */
void hpsb_iso_shutdown(struct hpsb_iso *iso);

-/* queue a packet for transmission. 'offset' is relative to the beginning of the
– DMA buffer, where the packet's data payload should already have been placed */
-int hpsb_iso_xmit_queue_packet(struct hpsb_iso *iso, u32 offset, u16 len, u8 tag, u8 sy);
+/* queue a packet for transmission.
+ * 'offset' is relative to the beginning of the DMA buffer, where the packet's
+ * data payload should already have been placed. */
+int hpsb_iso_xmit_queue_packet(struct hpsb_iso *iso, u32 offset, u16 len,

```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
+ u8 tag, u8 sy);

/* wait until all queued packets have been transmitted to the bus */
int hpsb_iso_xmit_sync(struct hpsb_iso *iso);

/* N packets have been read out of the buffer, re-use the buffer space */
-int hpsb_iso_rcv_release_packets(struct hpsb_iso *rcv, unsigned int n_packets);
+int hpsb_iso_rcv_release_packets(struct hpsb_iso *rcv,
+ unsigned int n_packets);

/* check for arrival of new packets immediately (even if irq_interval
- has not yet been reached) */
+ * has not yet been reached) */
int hpsb_iso_rcv_flush(struct hpsb_iso *iso);

/* returns # of packets ready to send or receive */
@@ -197,14 +202,15 @@ int hpsb_iso_n_ready(struct hpsb_iso *is
/* the following are callbacks available to low-level drivers */

/* call after a packet has been transmitted to the bus (interrupt context is OK)
- 'cycle' is the _exact_ cycle the packet was sent on
- 'error' should be non-zero if some sort of error occurred when sending the packet
-*/
+ * 'cycle' is the _exact_ cycle the packet was sent on
+ * 'error' should be non-zero if some sort of error occurred when sending the
+ * packet */
void hpsb_iso_packet_sent(struct hpsb_iso *iso, int cycle, int error);

/* call after a packet has been received (interrupt context OK) */
void hpsb_iso_packet_received(struct hpsb_iso *iso, u32 offset, u16 len,
- u16 total_len, u16 cycle, u8 channel, u8 tag, u8 sy);
+ u16 total_len, u16 cycle, u8 channel, u8 tag,
+ u8 sy);

/* call to wake waiting processes after buffer space has opened up. */
void hpsb_iso_wake(struct hpsb_iso *iso);
Index: linux/drivers/ieee1394/nodemgr.h
=====
--- linux.orig/drivers/ieee1394/nodemgr.h 2006-07-02 12:02:06.000000000 +0200
+++ linux/drivers/ieee1394/nodemgr.h 2006-07-02 12:16:52.000000000 +0200
@@ -44,7 +44,6 @@ struct bus_options {
u16 max_rec; /* Maximum packet size node can receive */
};

-
#define UNIT_DIRECTORY_VENDOR_ID 0x01
#define UNIT_DIRECTORY_MODEL_ID 0x02
#define UNIT_DIRECTORY_SPECIFIER_ID 0x04
@@ -59,8 +58,8 @@ struct bus_options {
* unit directory for each of these protocols.
*/
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
struct unit_directory {
- struct node_entry *ne; /* The node which this directory belongs to */
- octlet_t address; /* Address of the unit directory on the node */
+ struct node_entry *ne; /* The node which this directory belongs to */
+ octlet_t address; /* Address of the unit directory on the node */
u8 flags; /* Indicates which entries were read */

quadlet_t vendor_id;
@@ -79,11 +78,10 @@ struct unit_directory {
int length; /* Number of quadlets */

struct device device;
-
struct class_device class_dev;

struct csr1212_keyval *ud_kv;
- u32 lun; /* logical unit number immediate value */
+ u32 lun; /* logical unit number immediate value */
};

struct node_entry {
@@ -106,7 +104,6 @@ struct node_entry {
struct hpsb_tlabel_pool *tpool;

struct device device;
-
struct class_device class_dev;

/* Means this node is not attached anymore */
@@ -153,8 +150,8 @@ static inline int hpsb_node_entry_valid(
/*
* This will fill in the given, pre-initialised hpsb_packet with the current
* information from the node entry (host, node ID, generation number). It will
- * return false if the node owning the GUID is not accessible (and not modify the
- * hpsb_packet) and return true otherwise.
+ * return false if the node owning the GUID is not accessible (and not modify
+ * the hpsb_packet) and return true otherwise.
*
* Note that packet sending may still fail in hpsb_send_packet if a bus reset
* happens while you are trying to set up the packet (due to obsolete generation
@@ -170,16 +167,13 @@ int hpsb_node_write(struct node_entry *n
int hpsb_node_lock(struct node_entry *ne, u64 addr,
int extcode, quadlet_t *data, quadlet_t arg);

-
/* Iterate the hosts, calling a given function with supplied data for each
* host. */
int nodemgr_for_each_host(void *__data, int (*cb)(struct hpsb_host *, void *));

-
int init_ieee1394_nodemgr(void);
```

[PATCH 06/19] ieee1394: coding style and comment fixes in midlayer header files

```
void cleanup_ieee1394_nodemgr(void);
```

```
—
```

```
/* The template for a host device */
```

```
extern struct device nodemgr_dev_template_host;
```

```
—
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>