

[PATCH 5/8] Char: moxa, remove unused functions

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-12/msg08064.html>

- *From:* Jiri Slaby <jirislaby@xxxxxxxxxx>
 - *Date:* Sun, 31 Dec 2006 02:05:00 +0100 (CET)
-

moxa, remove unused functions

Remove ifdeffed functions and cleanup comments including too long license terms.

Signed-off-by: Jiri Slaby <jirislaby@xxxxxxxxxx>

commit c2eee5df210da17dfdea909b89f5db31b577f92a
tree cace45a59508ef209c42e662836ba861ecd06e81
parent 9146480faf6469f789229e4f09d99a90b3e05c26
author Jiri Slaby <jirislaby@xxxxxxxxxx> Sun, 31 Dec 2006 01:50:23 +0059
committer Jiri Slaby <jirislaby@xxxxxxxxxx> Sun, 31 Dec 2006 01:50:23 +0059

drivers/char/moxa.c | 245 -----
1 files changed, 0 insertions(+), 245 deletions(-)

```
diff --git a/drivers/char/moxa.c b/drivers/char/moxa.c
index ef2558f..3c8858c 100644
--- a/drivers/char/moxa.c
+++ b/drivers/char/moxa.c
@@ -11,15 +11,6 @@
 * 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., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

/*
@@ -1684,9 +1675,7 @@ int MoxaPortsOfCard(int cardno)
 * 2. MoxaPortEnable(int port); *
```

[PATCH 5/8] Char: moxa, remove unused functions

```
* 3. MoxaPortDisable(int port); *
* 4. MoxaPortGetMaxBaud(int port); *
- * 5. MoxaPortGetCurBaud(int port); *
* 6. MoxaPortSetBaud(int port, long baud); *
- * 7. MoxaPortSetMode(int port, int databit, int stopbit, int parity); *
* 8. MoxaPortSetTermio(int port, unsigned char *termio); *
* 9. MoxaPortGetLineOut(int port, int *dtrState, int *rtsState); *
* 10. MoxaPortLineCtrl(int port, int dtrState, int rtsState); *
@@ -1697,18 +1686,12 @@ int MoxaPortsOfCard(int cardno)
* 15. MoxaPortFlushData(int port, int mode); *
* 16. MoxaPortWriteData(int port, unsigned char * buffer, int length); *
* 17. MoxaPortReadData(int port, struct tty_struct *tty); *
- * 18. MoxaPortTxBufSize(int port); *
- * 19. MoxaPortRxBufSize(int port); *
* 20. MoxaPortTxQueue(int port); *
* 21. MoxaPortTxFree(int port); *
* 22. MoxaPortRxQueue(int port); *
- * 23. MoxaPortRxFree(int port); *
* 24. MoxaPortTxDisable(int port); *
* 25. MoxaPortTxEnable(int port); *
- * 26. MoxaPortGetBrkCnt(int port); *
* 27. MoxaPortResetBrkCnt(int port); *
- * 28. MoxaPortSetXonXoff(int port, int xonValue, int xoffValue); *
- * 29. MoxaPortIsTxHold(int port); *
* 30. MoxaPortSendBreak(int port, int ticks); *
*****/
/*
@@ -1795,15 +1778,6 @@ int MoxaPortsOfCard(int cardno)
* 38400/57600/115200 bps
*
*
- * Function 9: Get the current baud rate of this port.
- * Syntax:
- * long MoxaPortGetCurBaud(int port);
- * int port : port number (0 - 127)
- *
- * return: 0 : this port is invalid
- * 50 - 115200 bps
- *
- *
* Function 10: Setting baud rate of this port.
* Syntax:
* long MoxaPortSetBaud(int port, long baud);
@@ -1817,18 +1791,6 @@ int MoxaPortsOfCard(int cardno)
* baud rate will be the maximum baud rate.
*
*
- * Function 11: Setting the data-bits/stop-bits/parity of this port
- * Syntax:
- * int MoxaPortSetMode(int port, int databits, int stopbits, int parity);
- * int port : port number (0 - 127)
```

[PATCH 5/8] Char: moxa, remove unused functions

```
– * int databits : data bits (8/7/6/5)
– * int stopbits : stop bits (2/1/0, 0 show 1.5 stop bits)
– int parity : parity (0:None,1:Odd,2:Even,3:Mark,4:Space)
– *
– * return: –1 : invalid parameter
– * 0 : setting O.K.
– *
– *
* Function 12: Configure the port.
* Syntax:
* int MoxaPortSetTermio(int port, struct ktermios *termio, speed_t baud);
@@ –1933,22 +1895,6 @@ int MoxaPortsOfCard(int cardno)
* return: 0 – length : real read data length
*
*
– * Function 22: Get the Tx buffer size of this port
– * Syntax:
– * int MoxaPortTxBufSize(int port);
– * int port : port number (0 – 127)
– *
– * return: .. : Tx buffer size
– *
– *
– * Function 23: Get the Rx buffer size of this port
– * Syntax:
– * int MoxaPortRxBufSize(int port);
– * int port : port number (0 – 127)
– *
– * return: .. : Rx buffer size
– *
– *
* Function 24: Get the Tx buffer current queued data bytes
* Syntax:
* int MoxaPortTxQueue(int port);
@@ –1973,14 +1919,6 @@ int MoxaPortsOfCard(int cardno)
* return: .. : Rx buffer current queued data bytes
*
*
– * Function 27: Get the Rx buffer current free space
– * Syntax:
– * int MoxaPortRxFree(int port);
– * int port : port number (0 – 127)
– *
– * return: .. : Rx buffer current free space
– *
– *
* Function 28: Disable port data transmission.
* Syntax:
* void MoxaPortTxDisable(int port);
@@ –1993,14 +1931,6 @@ int MoxaPortsOfCard(int cardno)
* int port : port number (0 – 127)
```

[PATCH 5/8] Char: moxa, remove unused functions

```
*
*
- * Function 30: Get the received BREAK signal count.
- * Syntax:
- * int MoxaPortGetBrkCnt(int port);
- * int port : port number (0 – 127)
- *
- * return: 0 – .. : BREAK signal count
- *
- *
* Function 31: Get the received BREAK signal count and reset it.
* Syntax:
* int MoxaPortResetBrkCnt(int port);
@@ -2009,25 +1939,6 @@ int MoxaPortsOfCard(int cardno)
* return: 0 – .. : BREAK signal count
*
*
- * Function 32: Set the S/W flow control new XON/XOFF value, default
- * XON is 0x11 & XOFF is 0x13.
- * Syntax:
- * void MoxaPortSetXonXoff(int port, int xonValue, int xoffValue);
- * int port : port number (0 – 127)
- * int xonValue : new XON value (0 – 255)
- * int xoffValue : new XOFF value (0 – 255)
- *
- *
- * Function 33: Check this port's transmission is hold by remote site
- * because the flow control.
- * Syntax:
- * int MoxaPortIsTxHold(int port);
- * int port : port number (0 – 127)
- *
- * return: 0 : normal
- * 1 : hold by remote site
- *
- *
* Function 34: Send out a BREAK signal.
* Syntax:
* void MoxaPortSendBreak(int port, int ms100);
@@ -2276,23 +2187,6 @@ int MoxaPortDCDON(int port)
return (n);
}

-
-/*
- int MoxaDumpMem(int port, unsigned char * buffer, int len)
- {
- int i;
- unsigned long baseAddr,ofsAddr,ofs;
-
- baseAddr = moxaBaseAddr[port / MAX_PORTS_PER_BOARD];
```

[PATCH 5/8] Char: moxa, remove unused functions

```
- ofs = baseAddr + DynPage_addr + pageofs;
- if (len > 0x2000L)
- len = 0x2000L;
- for (i = 0; i < len; i++)
- buffer[i] = readb(ofs+i);
- }
- */
-
-
int MoxaPortWriteData(int port, unsigned char * buffer, int len)
{
int c, total, i;
@@ -2944,16 +2838,6 @@ static int moxaloadc320(int cardno, void __iomem *baseAddr, int len, int
*numPor
return (0);
}

-#if 0
-long MoxaPortGetCurBaud(int port)
-{
-
- if (moxaChkPort[port] == 0)
- return (0);
- return (moxaCurBaud[port]);
-}
-#endif /* 0 */
-
static void MoxaSetFifo(int port, int enable)
{
void __iomem *ofsAddr = moxaTableAddr[port];
@@ -2966,132 +2850,3 @@ static void MoxaSetFifo(int port, int enable)
moxafunc(ofsAddr, FC_SetTxFIFOCnt, 16);
}
}
-
-#if 0
-int MoxaPortSetMode(int port, int databits, int stopbits, int parity)
-{
- void __iomem *ofsAddr;
- int val;
-
- val = 0;
- switch (databits) {
- case 5:
- val |= 0;
- break;
- case 6:
- val |= 1;
- break;
- case 7:
- val |= 2;
```

[PATCH 5/8] Char: moxa, remove unused functions

```
- break;
- case 8:
- val |= 3;
- break;
- default:
- return (-1);
- }
- switch (stopbits) {
- case 0:
- val |= 0;
- break; /* stop bits 1.5 */
- case 1:
- val |= 0;
- break;
- case 2:
- val |= 4;
- break;
- default:
- return (-1);
- }
- switch (parity) {
- case 0:
- val |= 0x00;
- break; /* None */
- case 1:
- val |= 0x08;
- break; /* Odd */
- case 2:
- val |= 0x18;
- break; /* Even */
- case 3:
- val |= 0x28;
- break; /* Mark */
- case 4:
- val |= 0x38;
- break; /* Space */
- default:
- return (-1);
- }
- ofsAddr = moxaTableAddr[port];
- moxafunc(ofsAddr, FC_SetMode, val);
- return (0);
-}
-
-int MoxaPortTxBufSize(int port)
- {
- void __iomem *ofsAddr;
- int size;
-
- ofsAddr = moxaTableAddr[port];
- size = readw(ofsAddr + TX_mask);
```

[PATCH 5/8] Char: moxa, remove unused functions

```
- return (size);
-}
-
-int MoxaPortRxBufSize(int port)
-{
- void __iomem *ofsAddr;
- int size;
-
- ofsAddr = moxaTableAddr[port];
- size = readw(ofsAddr + RX_mask);
- return (size);
-}
-
-int MoxaPortRxFree(int port)
-{
- void __iomem *ofsAddr;
- ushort rptr, wptr, mask;
- int len;
-
- ofsAddr = moxaTableAddr[port];
- rptr = readw(ofsAddr + RXrptr);
- wptr = readw(ofsAddr + RXwptr);
- mask = readw(ofsAddr + RX_mask);
- len = mask - ((wptr - rptr) & mask);
- return (len);
-}
-int MoxaPortGetBrkCnt(int port)
-{
- return (moxaBreakCnt[port]);
-}
-
-void MoxaPortSetXonXoff(int port, int xonValue, int xoffValue)
-{
- void __iomem *ofsAddr;
-
- ofsAddr = moxaTableAddr[port];
- writew(xonValue, ofsAddr + FuncArg);
- writew(xoffValue, ofsAddr + FuncArg1);
- writew(FC_SetXonXoff, ofsAddr + FuncCode);
- wait_finish(ofsAddr);
-}
-
-int MoxaPortIsTxHold(int port)
-{
- void __iomem *ofsAddr;
- int val;
-
- ofsAddr = moxaTableAddr[port];
- if ((moxa_boards[port / MAX_PORTS_PER_BOARD].boardType == MOXA_BOARD_C320_ISA) ||
- (moxa_boards[port / MAX_PORTS_PER_BOARD].boardType == MOXA_BOARD_C320_PCI)) {
- moxafunc(ofsAddr, FC_GetCCSR, 0);
```

[PATCH 5/8] Char: moxa, remove unused functions

```
- val = readw(ofsAddr + FuncArg);  
- if (val & 0x04)  
- return (1);  
- } else {  
- if (readw(ofsAddr + FlagStat) & Tx_flowOff)  
- return (1);  
- }  
- return (0);  
-}  
-#endif  
-
```

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/>