

[2.6 patch] drivers/char/rocket.c: make some code static

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-01/9337.html>

From: Adrian Bunk (*bunk_at_stusta.de*)

Date: 01/31/05

Date: Mon, 31 Jan 2005 18:33:33 +0100

To: tytso@mit.edu

This patch makes some needlessly global code static.

Signed-off-by: Adrian Bunk <bunk@stusta.de>

```
---
drivers/char/rocket.c      | 181 ++++++-----
drivers/char/rocket_int.h |  40 -----
2 files changed, 101 insertions(+), 120 deletions(-)
--- linux-2.6.11-rc2-mm2-full/drivers/char/rocket_int.h.old      2005-01-31 15:10:41.000000000 +0100
+++ linux-2.6.11-rc2-mm2-full/drivers/char/rocket_int.h        2005-01-31 15:20:56.000000000 +0100
@@ -1130,46 +1130,6 @@
 */
#define sWriteTxByte(IO,DATA) sOutB(IO,DATA)

-int sInitController(CONTROLLER_T * CtlP,
-                    int CtlNum,
-                    ByteIO_t MudbacIO,
-                    ByteIO_t * AiopIOList,
-                    int AiopIOListSize,
-                    int IRQNum, Byte_t Frequency, int PeriodicOnly);
-
-int sPCIInitController(CONTROLLER_T * CtlP,
-                       int CtlNum,
-                       ByteIO_t * AiopIOList,
-                       int AiopIOListSize,
-                       WordIO_t ConfigIO,
-                       int IRQNum,
-                       Byte_t Frequency,
-                       int PeriodicOnly,
-                       int altChanRingIndicator, int UPCIRingInd);
-
-int sReadAiopID(ByteIO_t io);
-int sReadAiopNumChan(WordIO_t io);
-int sInitChan(CONTROLLER_T * CtlP,
-              CHANNEL_T * ChP, int AiopNum, int ChanNum);
-Byte_t sGetRxErrStatus(CHANNEL_T * ChP);
-void sStopRxProcessor(CHANNEL_T * ChP);
-void sStopSWInFlowCtl(CHANNEL_T * ChP);
-void sFlushRxFIFO(CHANNEL_T * ChP);
-void sFlushTxFIFO(CHANNEL_T * ChP);
-int sWriteTxPrioByte(CHANNEL_T * ChP, Byte_t Data);
-void sEnInterrupts(CHANNEL_T * ChP, Word_t Flags);
-void sDisInterrupts(CHANNEL_T * ChP, Word_t Flags);
```

Linux-Kernel: [2.6 patch] drivers/char/rocket.c: make some code static

```
-void sModemReset(CONTROLLER_T * CtlP, int chan, int on);
-void sPCIModemReset(CONTROLLER_T * CtlP, int chan, int on);
-void sSetInterfaceMode(CHANNEL_T * ChP, Byte_t mode);
-
-extern Byte_t R[RDATASIZE];
-extern CONTROLLER_T sController[CTL_SIZE];
-extern Byte_t sIRQMap[16];
-extern Byte_t sBitMapClrTbl[8];
-extern Byte_t sBitMapSetTbl[8];
-extern int sClockPrescale;
-
/*
 * Begin Linux specific definitions for the Rocketport driver
 */
--- linux-2.6.11-rc2-mm2-full/drivers/char/rocket.c.old 2005-01-31 15:08:49.000000000 +0100
+++ linux-2.6.11-rc2-mm2-full/drivers/char/rocket.c      2005-01-31 15:33:26.000000000 +0100
@@ -169,6 +169,65 @@
 static unsigned char lineNumbers[MAX_RP_PORTS];
 static unsigned long nextLineNumber;

+static Byte_t RData[RDATASIZE] = {
+  0x00, 0x09, 0xf6, 0x82,
+  0x02, 0x09, 0x86, 0xfb,
+  0x04, 0x09, 0x00, 0x0a,
+  0x06, 0x09, 0x01, 0x0a,
+  0x08, 0x09, 0x8a, 0x13,
+  0x0a, 0x09, 0xc5, 0x11,
+  0x0c, 0x09, 0x86, 0x85,
+  0x0e, 0x09, 0x20, 0x0a,
+  0x10, 0x09, 0x21, 0x0a,
+  0x12, 0x09, 0x41, 0xff,
+  0x14, 0x09, 0x82, 0x00,
+  0x16, 0x09, 0x82, 0x7b,
+  0x18, 0x09, 0x8a, 0x7d,
+  0x1a, 0x09, 0x88, 0x81,
+  0x1c, 0x09, 0x86, 0x7a,
+  0x1e, 0x09, 0x84, 0x81,
+  0x20, 0x09, 0x82, 0x7c,
+  0x22, 0x09, 0x0a, 0x0a
+};
+
+static Byte_t RRegData[RREGDATASIZE] = {
+  0x00, 0x09, 0xf6, 0x82, /* 00: Stop Rx processor */
+  0x08, 0x09, 0x8a, 0x13, /* 04: Tx software flow control */
+  0x0a, 0x09, 0xc5, 0x11, /* 08: XON char */
+  0x0c, 0x09, 0x86, 0x85, /* 0c: XANY */
+  0x12, 0x09, 0x41, 0xff, /* 10: Rx mask char */
+  0x14, 0x09, 0x82, 0x00, /* 14: Compare/Ignore #0 */
+  0x16, 0x09, 0x82, 0x7b, /* 18: Compare #1 */
+  0x18, 0x09, 0x8a, 0x7d, /* 1c: Compare #2 */
+  0x1a, 0x09, 0x88, 0x81, /* 20: Interrupt #1 */
+  0x1c, 0x09, 0x86, 0x7a, /* 24: Ignore/Replace #1 */
+  0x1e, 0x09, 0x84, 0x81, /* 28: Interrupt #2 */
+  0x20, 0x09, 0x82, 0x7c, /* 2c: Ignore/Replace #2 */
+  0x22, 0x09, 0x0a, 0x0a /* 30: Rx FIFO Enable */
+};
+
+static CONTROLLER_T sController[CTL_SIZE] = {
+  {-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, {0, 0, 0, 0},
+   {0, 0, 0, 0}}, {-1, -1, -1, -1}, {0, 0, 0, 0}},
+  {-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, {0, 0, 0, 0},
+   {0, 0, 0, 0}}, {-1, -1, -1, -1}, {0, 0, 0, 0}},
```

Linux-Kernel: [2.6 patch] drivers/char/rocket.c: make some code static

```

+     {-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, {0, 0, 0, 0},
+      {0, 0, 0, 0}, {-1, -1, -1, -1}, {0, 0, 0, 0}},
+     {-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, {0, 0, 0, 0},
+      {0, 0, 0, 0}, {-1, -1, -1, -1}, {0, 0, 0, 0}}
+};
+
+static Byte_t sBitMapClrTbl[8] = {
+    0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f
+};
+
+static Byte_t sBitMapSetTbl[8] = {
+    0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80
+};
+
+static int sClockPrescale = 0x14;
+
+
+    /***** RocketPort Static Prototypes      *****/
+    static int __init init_ISA(int i);
+    static void rp_wait_until_sent(struct tty_struct *tty, int timeout);
@@ -177,6 +236,26 @@
+    static unsigned char GetLineNumber(int ctrl, int aiop, int ch);
+    static unsigned char SetLineNumber(int ctrl, int aiop, int ch);
+    static void rp_start(struct tty_struct *tty);
+static int sInitChan(CONTROLLER_T * CtlP, CHANNEL_T *ChP, int AiopNum,
+    int ChanNum);
+static void sSetInterfaceMode(CHANNEL_T * ChP, Byte_t mode);
+static void sFlushRxFIFO(CHANNEL_T * ChP);
+static void sFlushTxFIFO(CHANNEL_T * ChP);
+static void sEnInterrupts(CHANNEL_T * ChP, Word_t Flags);
+static void sDisInterrupts(CHANNEL_T * ChP, Word_t Flags);
+static void sModemReset(CONTROLLER_T * CtlP, int chan, int on);
+static void sPCIModemReset(CONTROLLER_T * CtlP, int chan, int on);
+static int sWriteTxPrioByte(CHANNEL_T * ChP, Byte_t Data);
+static int sPCIInitController(CONTROLLER_T * CtlP, int CtlNum,
+    ByteIO_t * AiopIOList, int AiopIOListSize,
+    WordIO_t ConfigIO, int IRQNum, Byte_t Frequency,
+    int PeriodicOnly, int altChanRingIndicator,
+    int UPCIRingInd);
+static int sInitController(CONTROLLER_T * CtlP, int CtlNum, ByteIO_t MudbacIO,
+    ByteIO_t * AiopIOList, int AiopIOListSize,
+    int IRQNum, Byte_t Frequency, int PeriodicOnly);
+static int sReadAiopID(ByteIO_t io);
+static int sReadAiopNumChan(WordIO_t io);
+
+    #ifndef MODULE
+    MODULE_AUTHOR("Theodore Ts'o");
@@ -1798,7 +1877,7 @@
+    *   init's aiopic and serial port hardware.
+    *   Inputs:  i is the board number (0-n)
+    */
-__init int register_PCI(int i, struct pci_dev *dev)
+static int __init register_PCI(int i, struct pci_dev *dev)
+    {
+        int num_aiops, aiop, max_num_aiops, num_chan, chan;
+        unsigned int aiopio[MAX_AIOPS_PER_BOARD];
@@ -2461,64 +2540,6 @@
+    #define FALSE 0
+    #endif
+
+static Byte_t RData[RDATASIZE] = {
+    0x00, 0x09, 0xf6, 0x82,

```

Linux-Kernel: [2.6 patch] drivers/char/rocket.c: make some code static

```

-     0x02, 0x09, 0x86, 0xfb,
-     0x04, 0x09, 0x00, 0x0a,
-     0x06, 0x09, 0x01, 0x0a,
-     0x08, 0x09, 0x8a, 0x13,
-     0x0a, 0x09, 0xc5, 0x11,
-     0x0c, 0x09, 0x86, 0x85,
-     0x0e, 0x09, 0x20, 0x0a,
-     0x10, 0x09, 0x21, 0x0a,
-     0x12, 0x09, 0x41, 0xff,
-     0x14, 0x09, 0x82, 0x00,
-     0x16, 0x09, 0x82, 0x7b,
-     0x18, 0x09, 0x8a, 0x7d,
-     0x1a, 0x09, 0x88, 0x81,
-     0x1c, 0x09, 0x86, 0x7a,
-     0x1e, 0x09, 0x84, 0x81,
-     0x20, 0x09, 0x82, 0x7c,
-     0x22, 0x09, 0x0a, 0x0a
-};
-
-static Byte_t RRegData[RREGDATASIZE] = {
-     0x00, 0x09, 0xf6, 0x82, /* 00: Stop Rx processor */
-     0x08, 0x09, 0x8a, 0x13, /* 04: Tx software flow control */
-     0x0a, 0x09, 0xc5, 0x11, /* 08: XON char */
-     0x0c, 0x09, 0x86, 0x85, /* 0c: XANY */
-     0x12, 0x09, 0x41, 0xff, /* 10: Rx mask char */
-     0x14, 0x09, 0x82, 0x00, /* 14: Compare/Ignore #0 */
-     0x16, 0x09, 0x82, 0x7b, /* 18: Compare #1 */
-     0x18, 0x09, 0x8a, 0x7d, /* 1c: Compare #2 */
-     0x1a, 0x09, 0x88, 0x81, /* 20: Interrupt #1 */
-     0x1c, 0x09, 0x86, 0x7a, /* 24: Ignore/Replace #1 */
-     0x1e, 0x09, 0x84, 0x81, /* 28: Interrupt #2 */
-     0x20, 0x09, 0x82, 0x7c, /* 2c: Ignore/Replace #2 */
-     0x22, 0x09, 0x0a, 0x0a /* 30: Rx FIFO Enable */
-};
-
-CONTROLLER_T sController[CTL_SIZE] = {
-     {-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, {0, 0, 0, 0},
-     {0, 0, 0, 0}, {-1, -1, -1, -1}, {0, 0, 0, 0}},
-     {-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, {0, 0, 0, 0},
-     {0, 0, 0, 0}, {-1, -1, -1, -1}, {0, 0, 0, 0}},
-     {-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, {0, 0, 0, 0},
-     {0, 0, 0, 0}, {-1, -1, -1, -1}, {0, 0, 0, 0}},
-     {-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, {0, 0, 0, 0},
-     {0, 0, 0, 0}, {-1, -1, -1, -1}, {0, 0, 0, 0}}
-};
-
-Byte_t sBitMapClrTbl[8] = {
-     0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf, 0xbf, 0x7f
-};
-
-Byte_t sBitMapSetTbl[8] = {
-     0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80
-};
-
-int sClockPrescale = 0x14;
-
-
-*****
Function: sInitController
Purpose: Initialization of controller global registers and controller
@@ -2589,9 +2610,9 @@
After this function all AIOPs on the controller are disabled,
they can be enabled with sEnAiop().

```

Linux–Kernel: [2.6 patch] drivers/char/rocket.c: make some code static

```

*/
-int sInitController(CONTROLLER_T * CtlP, int CtlNum, ByteIO_t MudbacIO,
-                   ByteIO_t * AiopIOList, int AiopIOListSize, int IRQNum,
-                   Byte_t Frequency, int PeriodicOnly)
+static int sInitController(CONTROLLER_T * CtlP, int CtlNum, ByteIO_t MudbacIO,
+                           ByteIO_t * AiopIOList, int AiopIOListSize,
+                           int IRQNum, Byte_t Frequency, int PeriodicOnly)
{
    int i;
    ByteIO_t io;
@@ -2722,11 +2743,11 @@
    After this function all AIOPs on the controller are disabled,
    they can be enabled with sEnAiop().
*/
-int sPCIIInitController(CONTROLLER_T * CtlP, int CtlNum,
-                       ByteIO_t * AiopIOList, int AiopIOListSize,
-                       WordIO_t ConfigIO, int IRQNum, Byte_t Frequency,
-                       int PeriodicOnly, int altChanRingIndicator,
-                       int UPCIRingInd)
+static int sPCIIInitController(CONTROLLER_T * CtlP, int CtlNum,
+                               ByteIO_t * AiopIOList, int AiopIOListSize,
+                               WordIO_t ConfigIO, int IRQNum, Byte_t Frequency,
+                               int PeriodicOnly, int altChanRingIndicator,
+                               int UPCIRingInd)
{
    int i;
    ByteIO_t io;
@@ -2784,7 +2805,7 @@
    Warnings: No context switches are allowed while executing this function.
*/
-int sReadAiopID(ByteIO_t io)
+static int sReadAiopID(ByteIO_t io)
{
    Byte_t AiopID;          /* ID byte from AIOP */
@@ -2810,7 +2831,7 @@
    AIOP, otherwise it is an 8 channel.
    Warnings: No context switches are allowed while executing this function.
*/
-int sReadAiopNumChan(WordIO_t io)
+static int sReadAiopNumChan(WordIO_t io)
{
    Word_t x;
    static Byte_t R[4] = { 0x00, 0x00, 0x34, 0x12 };
@@ -2841,8 +2862,8 @@
    No context switches are allowed while executing this function.
*/
-int sInitChan(CONTROLLER_T * CtlP, CHANNEL_T * ChP, int AiopNum,
-             int ChanNum)
+static int sInitChan(CONTROLLER_T * CtlP, CHANNEL_T * ChP, int AiopNum,
+                    int ChanNum)
{
    int i;
    WordIO_t AiopIO;
@@ -2989,7 +3010,7 @@
    After calling this function a delay of 4 uS is required to ensure
    that the receive processor is no longer processing this channel.
*/
-void sStopRxProcessor(CHANNEL_T * ChP)
+static void sStopRxProcessor(CHANNEL_T * ChP)

```

Linux-Kernel: [2.6 patch] drivers/char/rocket.c: make some code static

```

{
    Byte_t R[4];

@@ -3014,7 +3035,7 @@
    this function.
    Warnings: No context switches are allowed while executing this function.
    */
-void sFlushRxFIFO(CHANNEL_T * ChP)
+static void sFlushRxFIFO(CHANNEL_T * ChP)
{
    int i;
    Byte_t Ch;          /* channel number within AIOP */
@@ -3056,7 +3077,7 @@
    this function.
    Warnings: No context switches are allowed while executing this function.
    */
-void sFlushTxFIFO(CHANNEL_T * ChP)
+static void sFlushTxFIFO(CHANNEL_T * ChP)
{
    int i;
    Byte_t Ch;          /* channel number within AIOP */
@@ -3096,7 +3117,7 @@

    Warnings: No context switches are allowed while executing this function.
    */
-int sWriteTxPrioByte(CHANNEL_T * ChP, Byte_t Data)
+static int sWriteTxPrioByte(CHANNEL_T * ChP, Byte_t Data)
{
    Byte_t DWBuf[4];    /* buffer for double word writes */
    Word_t *WordPtr;    /* must be far because Win SS != DS */
@@ -3158,7 +3179,7 @@
    enable channel interrupts. This would allow the global interrupt
    status register to be used to determine which AIOPs need service.
    */
-void sEnInterrupts(CHANNEL_T * ChP, Word_t Flags)
+static void sEnInterrupts(CHANNEL_T * ChP, Word_t Flags)
{
    Byte_t Mask;        /* Interrupt Mask Register */

@@ -3202,7 +3223,7 @@
    this channel's bit from being set in the AIOP's Interrupt Channel
    Register.
    */
-void sDisInterrupts(CHANNEL_T * ChP, Word_t Flags)
+static void sDisInterrupts(CHANNEL_T * ChP, Word_t Flags)
{
    Byte_t Mask;        /* Interrupt Mask Register */

@@ -3218,7 +3239,7 @@
}

-void sSetInterfaceMode(CHANNEL_T * ChP, Byte_t mode)
+static void sSetInterfaceMode(CHANNEL_T * ChP, Byte_t mode)
{
    sOutB(ChP->CtlP->AiopIO[2], (mode & 0x18) | ChP->ChanNum);
}
@@ -3227,7 +3248,7 @@
    * Not an official SSCI function, but how to reset RocketModems.
    * ISA bus version
    */
-void sModemReset(CONTROLLER_T * CtlP, int chan, int on)

```

Linux-Kernel: [2.6 patch] drivers/char/rocket.c: make some code static

```
+static void sModemReset(CONTROLLER_T * CtlP, int chan, int on)
{
    ByteIO_t addr;
    Byte_t val;
@@ -3252,7 +3273,7 @@
 * Not an official SSCI function, but how to reset RocketModems.
 * PCI bus version
 */
-void sPCIModemReset(CONTROLLER_T * CtlP, int chan, int on)
+static void sPCIModemReset(CONTROLLER_T * CtlP, int chan, int on)
{
    ByteIO_t addr;
```

-

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

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

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