```c
/*
 * (C) Copyright 2002
 * Gary Jennejohn, DENX Software Engineering, <gj@denx.de>
 *
 * This program 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
 *
 */

#include <common.h>
#if defined(CONFIG_S3C2400) || defined(CONFIG_TRAB)
#include <s3c2400.h>
#elif defined(CONFIG_S3C2410)
#include <s3c2410.h>
#endif

/* ghcstop */
/*
include/s3c2410.h
typedef enum {
        S3C24X0_UART0,
        S3C24X0_UART1,
        S3C24X0_UART2
} S3C24X0_UARTS_NR;
으로 선언되어 있고,

include/configs/smdk2410.h
에
#define CONFIG_SERIAL1          1       // we use SERIAL 1 on SMDK2410
로 선언되어 있으므로 이게 사용된다.
*/
#ifdef CONFIG_SERIAL1
#define UART_NR S3C24X0_UART0

#elif CONFIG_SERIAL2
# if defined(CONFIG_TRAB)
#  #error "TRAB supports only CONFIG_SERIAL1"
# endif
#define UART_NR S3C24X0_UART1

#elif CONFIG_SERIAL3
# if defined(CONFIG_TRAB)
#  #error "TRAB supports only CONFIG_SERIAL1"
# endif
#define UART_NR S3C24X0_UART2

#else
#error "Bad: you didn't configure serial ..."
#endif

/* ghcstop */
/* 자세한 것은 메뉴얼을 보면 되겠는데...일단 무지 간단하네...나중에 쓸것^^ */
void serial_setbrg (void)
{
    DECLARE_GLOBAL_DATA_PTR;

    /*
     * UART_NR: 위의 설명을 보시요
     * : 세개의 UART중 해당 UART의 BASE address를 가져온다...여기서는 UART0다
     */
    S3C24X0_UART * const uart = S3C24X0_GetBase_UART(UART_NR);
    int i;
    unsigned int reg = 0;

    /*
     * include/configs/smdk2410.h에
     * #define CONFIG_BAUDRATE      115200
     * 으로 선언되어 있고, 이것은 환경변수 설정할때 default_environment[]에
     * macro로 들어가서 설정이 된다...아니면 environment.c의 environment[]에 들어갈거다..
     * 이에 관련된것은 env_init()쪽을 볼 것.
     */
```

```
83      /* value is calculated so : (int)(PCLK/16./baudrate) -1 */
84      reg = get_PCLK() / (16 * gd->baudrate) - 1;
85
86      /* FIFO enable, Tx/Rx FIFO clear */
87      uart->UFCON = 0x07;
88      uart->UMCON = 0x0;
89      /* Normal,No parity,1 stop,8 bit */
90      uart->ULCON = 0x3;
91      /*
92       * tx=level,rx=edge,disable timeout int.,enable rx error int.,
93       * normal,interrupt or polling
94       */
95      uart->UCON = 0x245;
96      uart->UBRDIV = reg;
97
98  /* hw flow control을 사용할 경우? */
99  #ifdef CONFIG_HWFLOW
100     uart->UMCON = 0x1; /* RTS up */
101 #endif
102     /* 설정한 뒤에 약간의 delay를 줘야하나 보다.,.,. */
103     for (i = 0; i < 100; i++);
104 }
105
106 /* ghcstop */
107 /*
108  * Initialise the serial port with the given baudrate. The settings
109  * are always 8 data bits, no parity, 1 stop bit, no start bits.
110  *
111  called in board.c
112  주어진 baudrate에 알맞게 시리얼 포트를 초기화. 세팅은 항상
113  8 data bits, no parity, 1 stop bit, no start bits 이렇게 된다.
114  */
115 int serial_init (void)
116 {
117     serial_setbrg ();
118
119     return (0);
120 }
121
122 /*
123  * Read a single byte from the serial port. Returns 1 on success, 0
124  * otherwise. When the function is succesfull, the character read is
125  * written into its argument c.
126  */
127 int serial_getc (void)
128 {
129     S3C24X0_UART * const uart = S3C24X0_GetBase_UART(UART_NR);
130
131     /* wait for character to arrive */
132     while (!(uart->UTRSTAT & 0x1));
133
134     return uart->URXH & 0xff;
135 }
136
137 #ifdef CONFIG_HWFLOW
138 static int hwflow = 0; /* turned off by default */
139 int hwflow_onoff(int on)
140 {
141     switch(on) {
142     case 0:
143     default:
144         break; /* return current */
145     case 1:
146         hwflow = 1; /* turn on */
147         break;
148     case -1:
149         hwflow = 0; /* turn off */
150         break;
151     }
152     return hwflow;
153 }
154 #endif
155
156 #ifdef CONFIG_MODEM_SUPPORT
157 static int be_quiet = 0;
158 void disable_putc(void)
159 {
160     be_quiet = 1;
161 }
162
163 void enable_putc(void)
164 {
```

```
165     be_quiet = 0;
166 }
167 #endif
168
169
170 /*
171  * Output a single byte to the serial port.
172  */
173 void serial_putc (const char c)
174 {
175     S3C24X0_UART * const uart = S3C24X0_GetBase_UART(UART_NR);
176 #ifdef CONFIG_MODEM_SUPPORT
177     if (be_quiet)
178         return;
179 #endif
180
181     /* wait for room in the tx FIFO */
182     while (!(uart->UTRSTAT & 0x2));
183
184 #ifdef CONFIG_HWFLOW
185     /* Wait for CTS up */
186     while(hwflow && !(uart->UMSTAT & 0x1))
187         ;
188 #endif
189
190     uart->UTXH = c;
191
192     /* If \n, also do \r */
193     if (c == '\n')
194         serial_putc ('\r');
195 }
196
197 /*
198  * Test whether a character is in the RX buffer
199  */
200 int serial_tstc (void)
201 {
202     S3C24X0_UART * const uart = S3C24X0_GetBase_UART(UART_NR);
203
204     return uart->UTRSTAT & 0x1;
205 }
206
207 void
208 serial_puts (const char *s)
209 {
210     while (*s) {
211         serial_putc (*s++);
212     }
213 }
214
```