```c
/*
 * (C) Copyright 2000-2002
 * Wolfgang Denk, DENX Software Engineering, wd@denx.de.
 *
 * (C) Copyright 2001 Sysgo Real-Time Solutions, GmbH <www.elinos.com>
 * Andreas Heppel <aheppel@sysgo.de>

 * See file CREDITS for list of people who contributed to this
 * project.
 *
 * 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
 */

/* #define DEBUG */

#include <common.h>

#if defined(CFG_ENV_IS_IN_FLASH) /* Environment is in Flash */

#include <command.h>
#include <environment.h>
#include <cmd_nvedit.h>
#include <linux/stddef.h>
#include <malloc.h>

#if ((CONFIG_COMMANDS&(CFG_CMD_ENV|CFG_CMD_FLASH)) == (CFG_CMD_ENV|CFG_CMD_FLASH))
#define CMD_SAVEENV
#elif defined(CFG_ENV_ADDR_REDUND)
#error Cannot use CFG_ENV_ADDR_REDUND without CFG_CMD_ENV & CFG_CMD_FLASH
#endif

#if defined(CFG_ENV_SIZE_REDUND) && (CFG_ENV_SIZE_REDUND < CFG_ENV_SIZE)
#error CFG_ENV_SIZE_REDUND should not be less then CFG_ENV_SIZE
#endif

#ifdef CONFIG_INFERNO
# ifdef CFG_ENV_ADDR_REDUND
#error CFG_ENV_ADDR_REDUND is not implemented for CONFIG_INFERNO
# endif
#endif

char * env_name_spec = "Flash";

#ifdef ENV_IS_EMBEDDED

extern uchar environment[];
env_t *env_ptr = (env_t *)(&environment[0]);

#ifdef CMD_SAVEENV
/* static env_t *flash_addr = (env_t *)(&environment[0]);-broken on ARM-wd-*/
static env_t *flash_addr = (env_t *)CFG_ENV_ADDR;
#endif

#else /* ! ENV_IS_EMBEDDED */

/* ghcstop */
/* ENV_IS_EMBEDDED이 선언이 안되어 있으므로....이렇게 사용한다...*/
env_t *env_ptr = (env_t *)CFG_ENV_ADDR;
#ifdef CMD_SAVEENV
static env_t *flash_addr = (env_t *)CFG_ENV_ADDR;
#endif

#endif /* ENV_IS_EMBEDDED */

#ifdef CFG_ENV_ADDR_REDUND
static env_t *flash_addr_new = (env_t *)CFG_ENV_ADDR_REDUND;

/* CFG_ENV_ADDR is supposed to be on sector boundary */
static ulong end_addr = CFG_ENV_ADDR + CFG_ENV_SECT_SIZE - 1;
```

```
 83 static ulong end_addr_new = CFG_ENV_ADDR_REDUND + CFG_ENV_SECT_SIZE - 1;
 84
 85 static uchar active_flag = 1;
 86 static uchar obsolete_flag = 0;
 87 #endif
 88
 89 extern uchar default_environment[];
 90 extern int default_environment_size;
 91
 92
 93 uchar env_get_char_spec (int index)
 94 {
 95     DECLARE_GLOBAL_DATA_PTR;
 96
 97     return ( *((uchar *)(gd->env_addr + index)) );
 98 }
 99
100 #ifdef CFG_ENV_ADDR_REDUND
101
102 int  env_init(void)
103 {
104     DECLARE_GLOBAL_DATA_PTR;
105
106     int crc1_ok =
107         (crc32(0, flash_addr->data, ENV_SIZE) == flash_addr->crc);
108     int crc2_ok =
109         (crc32(0, flash_addr_new->data, ENV_SIZE) == flash_addr_new->crc);
110
111     uchar flag1 = flash_addr->flags;
112     uchar flag2 = flash_addr_new->flags;
113
114     ulong addr_default = (ulong)&default_environment[0];
115     ulong addr1 = (ulong)&(flash_addr->data);
116     ulong addr2 = (ulong)&(flash_addr_new->data);
117
118     if (crc1_ok && ! crc2_ok)
119     {
120         gd->env_addr  = addr1;
121         gd->env_valid = 1;
122     }
123     else if (! crc1_ok && crc2_ok)
124     {
125         gd->env_addr  = addr2;
126         gd->env_valid = 1;
127     }
128     else if (! crc1_ok && ! crc2_ok)
129     {
130         gd->env_addr  = addr_default;
131         gd->env_valid = 0;
132     }
133     else if (flag1 == active_flag && flag2 == obsolete_flag)
134     {
135         gd->env_addr  = addr1;
136         gd->env_valid = 1;
137     }
138     else if (flag1 == obsolete_flag && flag2 == active_flag)
139     {
140         gd->env_addr  = addr2;
141         gd->env_valid = 1;
142     }
143     else if (flag1 == flag2)
144     {
145         gd->env_addr  = addr1;
146         gd->env_valid = 2;
147     }
148     else if (flag1 == 0xFF)
149     {
150         gd->env_addr  = addr1;
151         gd->env_valid = 2;
152     }
153     else if (flag2 == 0xFF)
154     {
155         gd->env_addr  = addr2;
156         gd->env_valid = 2;
157     }
158
159     return (0);
160 }
161
162 #ifdef CMD_SAVEENV
163 int saveenv(void)
164 {
```

```
165     char *saved_data = NULL;
166     int rc = 1;
167 #if CFG_ENV_SECT_SIZE > CFG_ENV_SIZE
168     ulong up_data = 0;
169 #endif
170 
171     debug ("Protect off %08lX ... %08lX\n",
172         (ulong)flash_addr, end_addr);
173 
174     if (flash_sect_protect (0, (ulong)flash_addr, end_addr)) {
175         goto Done;
176     }
177 
178     debug ("Protect off %08lX ... %08lX\n",
179         (ulong)flash_addr_new, end_addr_new);
180 
181     if (flash_sect_protect (0, (ulong)flash_addr_new, end_addr_new)) {
182         goto Done;
183     }
184 
185 #if CFG_ENV_SECT_SIZE > CFG_ENV_SIZE
186     up_data = (end_addr_new + 1 - ((long)flash_addr_new + CFG_ENV_SIZE));
187     debug ("Data to save 0x%x\n", up_data);
188     if (up_data) {
189         if ((saved_data = malloc(up_data)) == NULL) {
190             printf("Unable to save the rest of sector (%ld)\n",
191                 up_data);
192             goto Done;
193         }
194         memcpy(saved_data,
195             (void *)((long)flash_addr_new + CFG_ENV_SIZE), up_data);
196         debug ("Data (start 0x%x, len 0x%x) saved at 0x%x\n",
197                (long)flash_addr_new + CFG_ENV_SIZE,
198                up_data, saved_data);
199     }
200 #endif
201     puts ("Erasing Flash...");
202     debug (" %08lX ... %08lX ...",
203         (ulong)flash_addr_new, end_addr_new);
204 
205     if (flash_sect_erase ((ulong)flash_addr_new, end_addr_new)) {
206         goto Done;
207     }
208 
209     puts ("Writing to Flash... ");
210     debug (" %08lX ... %08lX ...",
211         (ulong)&(flash_addr_new->data),
212         sizeof(env_ptr->data)+(ulong)&(flash_addr_new->data));
213     if (flash_write(env_ptr->data,
214                     (ulong)&(flash_addr_new->data),
215             sizeof(env_ptr->data)) ||
216 
217         flash_write((char *)&(env_ptr->crc),
218                     (ulong)&(flash_addr_new->crc),
219             sizeof(env_ptr->crc)) ||
220 
221         flash_write((char *)&obsolete_flag,
222                     (ulong)&(flash_addr->flags),
223             sizeof(flash_addr->flags)) ||
224 
225         flash_write((char *)&active_flag,
226                     (ulong)&(flash_addr_new->flags),
227             sizeof(flash_addr_new->flags)))
228     {
229         flash_perror (rc);
230         goto Done;
231     }
232     puts ("done\n");
233 
234 #if CFG_ENV_SECT_SIZE > CFG_ENV_SIZE
235     if (up_data) { /* restore the rest of sector */
236         debug ("Restoring the rest of data to 0x%x len 0x%x\n",
237                (long)flash_addr_new + CFG_ENV_SIZE, up_data);
238         if (flash_write(saved_data,
239                 (long)flash_addr_new + CFG_ENV_SIZE,
240                 up_data)) {
241             flash_perror(rc);
242             goto Done;
243         }
244     }
245 #endif
246     {
```

```
247         env_t * etmp = flash_addr;
248         ulong ltmp = end_addr;
249
250         flash_addr = flash_addr_new;
251         flash_addr_new = etmp;
252
253         end_addr = end_addr_new;
254         end_addr_new = ltmp;
255     }
256
257     rc = 0;
258 Done:
259
260     if (saved_data)
261         free (saved_data);
262     /* try to re-protect */
263     (void) flash_sect_protect (1, (ulong)flash_addr, end_addr);
264     (void) flash_sect_protect (1, (ulong)flash_addr_new, end_addr_new);
265
266     return rc;
267 }
268 #endif /* CMD_SAVEENV */
269
270 #else /* ! CFG_ENV_ADDR_REDUND */
271
272 /* ghcstop */
273 /*
274  * lib_arm/board.c에서 호출되는 함수이다.
275  */
276 int  env_init(void)
277 {
278     DECLARE_GLOBAL_DATA_PTR;
279
280     /*
281      * 계산된 데이터위치의 crc32값과 저장되어 있는 env_ptr->crc값과
282      * 일치하는지 안하는지 검사
283      */
284     if (crc32(0, env_ptr->data, ENV_SIZE) == env_ptr->crc) {
285         gd->env_addr  = (ulong)&(env_ptr->data);
286         gd->env_valid = 1;
287     }
288     else /* 환경변수를 write 안 했을 때, 즉, 맨처음은 여기 걸릴것 같다 */
289     {
290         /*
291          * common/env_common.c의 uchar default_environment[]
292          * 를 기본 환경변수로 세팅한다...^^
293          */
294         gd->env_addr  = (ulong)&default_environment[0];
295         /*
296          * 요건 나중에 common/env_common.c의 env_relocate()에서 재 계산후
297          * 1로 바뀌는 것 같음
298          */
299         gd->env_valid = 0;
300     }
301
302     return (0);
303 }
304
305 #ifdef CMD_SAVEENV
306
307 int saveenv(void)
308 {
309     int len, rc;
310     ulong   end_addr;
311     ulong   flash_sect_addr;
312 #if defined(CFG_ENV_SECT_SIZE) && (CFG_ENV_SECT_SIZE > CFG_ENV_SIZE)
313     ulong   flash_offset;
314     uchar   env_buffer[CFG_ENV_SECT_SIZE];
315 #else
316     uchar *env_buffer = (char *)env_ptr;
317 #endif  /* CFG_ENV_SECT_SIZE */
318     int rcode = 0;
319
320 #if defined(CFG_ENV_SECT_SIZE) && (CFG_ENV_SECT_SIZE > CFG_ENV_SIZE)
321
322     flash_offset    = ((ulong)flash_addr) & (CFG_ENV_SECT_SIZE-1);
323     flash_sect_addr = ((ulong)flash_addr) & ~(CFG_ENV_SECT_SIZE-1);
324
325     debug ( "copy old content: "
326         "sect_addr: %08lX  env_addr: %08lX  offset: %08lX\n",
327         flash_sect_addr, (ulong)flash_addr, flash_offset);
328
```

```
329     /* copy old contents to temporary buffer */
330     memcpy (env_buffer, (void *)flash_sect_addr, CFG_ENV_SECT_SIZE);
331
332     /* copy current environment to temporary buffer */
333     memcpy ((uchar *)((unsigned long)env_buffer + flash_offset),
334         env_ptr,
335         CFG_ENV_SIZE);
336
337     len  = CFG_ENV_SECT_SIZE;
338 #else
339     flash_sect_addr = (ulong)flash_addr;
340     len  = CFG_ENV_SIZE;
341 #endif  /* CFG_ENV_SECT_SIZE */
342
343 #ifndef CONFIG_INFERNO
344     end_addr = flash_sect_addr + len - 1;
345 #else
346     /* this is the last sector, and the size is hardcoded here */
347     /* otherwise we will get stack problems on loading 128 KB environment */
348     end_addr = flash_sect_addr + 0x20000 - 1;
349 #endif
350
351     debug ("Protect off %08lX ... %08lX\n",
352         (ulong)flash_sect_addr, end_addr);
353
354     if (flash_sect_protect (0, flash_sect_addr, end_addr))
355         return 1;
356
357     puts ("Erasing Flash...");
358     if (flash_sect_erase (flash_sect_addr, end_addr))
359         return 1;
360
361     puts ("Writing to Flash... ");
362     rc = flash_write(env_buffer, flash_sect_addr, len);
363     if (rc != 0) {
364         flash_perror (rc);
365         rcode = 1;
366     } else {
367         puts ("done\n");
368     }
369
370     /* try to re-protect */
371     (void) flash_sect_protect (1, flash_sect_addr, end_addr);
372     return rcode;
373 }
374
375 #endif /* CMD_SAVEENV */
376
377 #endif /* CFG_ENV_ADDR_REDUND */
378
379 void env_relocate_spec (void)
380 {
381 #if !defined(ENV_IS_EMBEDDED) || defined(CFG_ENV_ADDR_REDUND)
382 #ifdef CFG_ENV_ADDR_REDUND
383     DECLARE_GLOBAL_DATA_PTR;
384
385     if (gd->env_addr != (ulong)&(flash_addr->data))
386     {
387         env_t * etmp = flash_addr;
388         ulong ltmp = end_addr;
389
390         flash_addr = flash_addr_new;
391         flash_addr_new = etmp;
392
393         end_addr = end_addr_new;
394         end_addr_new = ltmp;
395     }
396
397     if (flash_addr_new->flags != obsolete_flag &&
398         crc32(0, flash_addr_new->data, ENV_SIZE) ==
399         flash_addr_new->crc)
400     {
401         gd->env_valid = 2;
402         flash_sect_protect (0, (ulong)flash_addr_new, end_addr_new);
403         flash_write((char *)&obsolete_flag,
404                             (ulong)&(flash_addr_new->flags),
405                     sizeof(flash_addr_new->flags));
406         flash_sect_protect (1, (ulong)flash_addr_new, end_addr_new);
407     }
408
409     if (flash_addr->flags != active_flag &&
410         (flash_addr->flags & active_flag) == active_flag)
```

```
411     {
412         gd->env_valid = 2;
413         flash_sect_protect (0, (ulong)flash_addr, end_addr);
414         flash_write((char *)&active_flag,
415                             (ulong)&(flash_addr->flags),
416                     sizeof(flash_addr->flags));
417         flash_sect_protect (1, (ulong)flash_addr, end_addr);
418     }
419
420     if (gd->env_valid == 2)
421         puts ("*** Warning - some problems detected "
422               "reading environment; recovered successfully\n\n");
423 #endif /* CFG_ENV_ADDR_REDUND */
424     memcpy (env_ptr, (void*)flash_addr, CFG_ENV_SIZE);
425 #endif /* ! ENV_IS_EMBEDDED || CFG_ENV_ADDR_REDUND */
426 }
427
428 #endif /* CFG_ENV_IS_IN_FLASH */
429
```