

```

1  /*
2  * (C) Copyright 2001
3  * Erik Theisen, Wave 7 Optics, etheisen@mindspring.com.
4  *
5  * See file CREDITS for list of people who contributed to this
6  * project.
7  *
8  * This program is free software; you can redistribute it and/or
9  * modify it under the terms of the GNU General Public License as
10 * published by the Free Software Foundation; either version 2 of
11 * the License, or (at your option) any later version.
12 *
13 * This program is distributed in the hope that it will be useful,
14 * but WITHOUT ANY WARRANTY; without even the implied warranty of
15 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 * GNU General Public License for more details.
17 *
18 * You should have received a copy of the GNU General Public License
19 * along with this program; if not, write to the Free Software
20 * Foundation, Inc., 59 Temple Place, Suite 330, Boston,
21 * MA 02111-1307 USA
22 */
23
24 #include <config.h>
25 #include <environment.h>
26
27 /*
28 * Handle HOSTS that have prepended
29 * crap on symbol names, not TARGETS.
30 */
31 #if defined(__APPLE__)
32 /* Leading underscore on symbols */
33 # define SYM_CHAR "_"
34 #else /* No leading character on symbols */
35 # define SYM_CHAR
36 #endif
37
38 /*
39 * Generate embedded environment table
40 * inside U-Boot image, if needed.
41 */
42 #if defined(ENV_IS_EMBEDDED)
43 /*
44 * Only put the environment in it's own section when we are building
45 * U-Boot proper. The host based program "tools/envcrc" does not need
46 * a seperate section. Note that ENV_CRC is only defined when building
47 * U-Boot itself.
48 */
49 #if (defined(CONFIG_CMI) || \
50      defined(CONFIG_FADS) || \
51      defined(CONFIG_HYMOD) || \
52      defined(CONFIG_ICU862) || \
53      defined(CONFIG_R360MPI) || \
54      defined(CONFIG_TQM8xxL) || \
55      defined(CONFIG_RRVISION) || \
56      defined(CONFIG_TRAB) ) && \
57      defined(ENV_CRC) /* Environment embedded in U-Boot .ppcenv section */
58 /* XXX - This only works with GNU C */
59 # define __PPCENV__ __attribute__((section(".ppcenv")))
60 # define __PPCTEXT__ __attribute__((section(".text")))
61
62 #elif defined(USE_HOSTCC) /* Native for 'tools/envcrc' */
63 # define __PPCENV__ /*XXX DO_NOT_DEL_THIS_COMMENT*/
64 # define __PPCTEXT__ /*XXX DO_NOT_DEL_THIS_COMMENT*/
65
66 #else /* Environment is embedded in U-Boot's .text section */
67 /* XXX - This only works with GNU C */
68 # define __PPCENV__ __attribute__((section(".text")))
69 # define __PPCTEXT__ __attribute__((section(".text")))
70 #endif
71
72 /*
73 * Macros to generate global absolutes.
74 */
75 #define GEN_SYMNAME(str) SYM_CHAR #str
76 #define GEN_VALUE(str) #str
77 #define GEN_ABS(name, value) \
78     asm (".globl " GEN_SYMNAME(name)); \
79     asm (GEN_SYMNAME(name) " = " GEN_VALUE(value))
80
81 /*
82 * Macros to transform values

```

```

83  * into environment strings.
84  */
85  #define XMK_STR(x)  #x
86  #define MK_STR(x)   XMK_STR(x)
87
88  /*
89   * Check to see if we are building with a
90   * computed CRC.  Otherwise define it as ~0.
91   */
92  #if !defined(ENV_CRC)
93  # define ENV_CRC  ~0
94  #endif
95
96  /* ghcstop */
97  /* include/environment.h
98  typedef struct environment_s {
99      unsigned long   crc;           // CRC32 over data bytes
100      unsigned char   data[ENV_SIZE]; // Environment data
101  } env_t;
102
103                                     (environment.data)      가
104  bootargs=root=ramfs devfs=mount console=ttySA0,9600"\0"
105  bootcmd=tftp; bootm"\0"
106
107      ....
108  ex>
109  #include <stdio.h>
110
111  unsigned char data[100] =
112  {
113      "bootargs="      "babo"          "\0"
114      "bootcmd="       "babol"        "\0"
115      "\0"
116  };
117
118  main()
119  {
120      printf("%s\n", data);
121      printf("%s\n", data+(strlen(data)+1) );
122  }
123
124  output>
125  bootargs=babo
126  bootcmd=babol
127  */
128
129  env_t environment __PPCENV__ = {
130      ENV_CRC,           /* CRC Sum */
131      #ifdef CFG_REDUNDAND_ENVIRONMENT
132          1,             /* Flags: valid */
133      #endif
134      {
135          #if defined(CONFIG_BOOTARGS)
136              "bootargs=" CONFIG_BOOTARGS          "\0"
137          #endif
138          #if defined(CONFIG_BOOTCOMMAND)
139              "bootcmd="  CONFIG_BOOTCOMMAND       "\0"
140          #endif
141          #if defined(CONFIG_RAMBOOTCOMMAND)
142              "ramboot="  CONFIG_RAMBOOTCOMMAND    "\0"
143          #endif
144          #if defined(CONFIG_NFSBOOTCOMMAND)
145              "nfsboot="  CONFIG_NFSBOOTCOMMAND    "\0"
146          #endif
147          #if defined(CONFIG_BOOTDELAY) && (CONFIG_BOOTDELAY >= 0)
148              "bootdelay=" MK_STR(CONFIG_BOOTDELAY) "\0"
149          #endif
150          #if defined(CONFIG_BAUDRATE) && (CONFIG_BAUDRATE >= 0)
151              "baudrate=" MK_STR(CONFIG_BAUDRATE)  "\0"
152          #endif
153          #ifdef CONFIG_LOADS_ECHO
154              "loads_echo=" MK_STR(CONFIG_LOADS_ECHO)  "\0"
155          #endif
156          #ifdef CONFIG_ETHADDR
157              "ethaddr="  MK_STR(CONFIG_ETHADDR)      "\0"
158          #endif
159          #ifdef CONFIG_ETH1ADDR
160              "eth1addr=" MK_STR(CONFIG_ETH1ADDR)     "\0"
161          #endif
162          #ifdef CONFIG_ETH2ADDR
163              "eth2addr=" MK_STR(CONFIG_ETH2ADDR)     "\0"
164          #endif

```

```
165 #ifdef CONFIG_ETHPRIME
166     "ethprime=" CONFIG_ETHPRIME        "\0"
167 #endif
168 #ifdef CONFIG_IPADDR
169     "ipaddr=" MK_STR(CONFIG_IPADDR)     "\0"
170 #endif
171 #ifdef CONFIG_SERVERIP
172     "serverip=" MK_STR(CONFIG_SERVERIP) "\0"
173 #endif
174 #ifdef CFG_AUTOLOAD
175     "autoload=" CFG_AUTOLOAD            "\0"
176 #endif
177 #ifdef CONFIG_ROOTPATH
178     "rootpath=" MK_STR(CONFIG_ROOTPATH) "\0"
179 #endif
180 #ifdef CONFIG_GATEWAYIP
181     "gatewayip=" MK_STR(CONFIG_GATEWAYIP) "\0"
182 #endif
183 #ifdef CONFIG_NETMASK
184     "netmask=" MK_STR(CONFIG_NETMASK)   "\0"
185 #endif
186 #ifdef CONFIG_HOSTNAME
187     "hostname=" MK_STR(CONFIG_HOSTNAME) "\0"
188 #endif
189 #ifdef CONFIG_BOOTFILE
190     "bootfile=" MK_STR(CONFIG_BOOTFILE) "\0"
191 #endif
192 #ifdef CONFIG_LOADADDR
193     "loadaddr=" MK_STR(CONFIG_LOADADDR) "\0"
194 #endif
195 #ifdef CONFIG_PREBOOT
196     "preboot=" CONFIG_PREBOOT           "\0"
197 #endif
198 #ifdef CONFIG_CLOCKS_IN_MHZ
199     "clocks_in_mhz=" "1"                "\0"
200 #endif
201 #if defined(CONFIG_PCI_BOOTDELAY) && (CONFIG_PCI_BOOTDELAY > 0)
202     "pcideLAY=" MK_STR(CONFIG_PCI_BOOTDELAY) "\0"
203 #endif
204 #ifdef CONFIG_EXTRA_ENV_SETTINGS
205     CONFIG_EXTRA_ENV_SETTINGS
206 #endif
207     "\0" /* Term. env_t.data with 2 NULs */
208 }
209 };
210 #ifdef CFG_ENV_ADDR_REDUND
211 env_t redundand_environment __PPCENV__ = {
212     0, /* CRC Sum: invalid */
213     0, /* Flags: invalid */
214     {
215         "\0"
216     }
217 };
218 #endif /* CFG_ENV_ADDR_REDUND */
219
220 /*
221  * These will end up in the .text section
222  * if the environment strings are embedded
223  * in the image. When this is used for
224  * tools/envcrc, they are placed in the
225  * .data/.sdata section.
226  */
227
228 unsigned long env_size __PPCTEXT__ = sizeof(env_t);
229
230 /*
231  * Add in absolutes.
232  */
233 GEN_ABS(env_offset, CFG_ENV_OFFSET);
234
235 #endif /* ENV_IS_EMBEDDED */
236
```