

```

1:  /*
2:  libxbee - a C library to aid the use of Digi's Series 1 XBee modules
3:           running in API mode (AP=2).
4:
5:  Copyright (C) 2009 Attie Grande (attie@attie.co.uk)
6:
7:  This program is free software: you can redistribute it and/or modify
8:  it under the terms of the GNU General Public License as published by
9:  the Free Software Foundation, either version 3 of the License, or
10: (at your option) any later version.
11:
12: This program is distributed in the hope that it will be useful,
13: but WITHOUT ANY WARRANTY; without even the implied warranty of
14: MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15: GNU General Public License for more details.
16:
17: You should have received a copy of the GNU General Public License
18: along with this program. If not, see <http://www.gnu.org/licenses/>.
19: */
20:
21: #include <stdio.h>
22: #include <stdlib.h>
23:
24: #include <stdarg.h>
25:
26: #include <string.h>
27: #include <fcntl.h>
28: #include <errno.h>
29: #include <signal.h>
30:
31: #ifdef __GNUC__ /* ---- */
32: #include <unistd.h>
33: #include <termios.h>
34: #define __USE_GNU
35: #include <pthread.h>
36: #undef __USE_GNU
37: #include <sys/time.h>
38: #else /* ----- */
39: #include <Windows.h>
40: #include <io.h>
41: #include <time.h>
42: #include <sys/timeb.h>
43: #endif /* ----- */
44:
45: #ifdef __UMAKEFILE
46: #define HOST_OS "Embedded"
47: #elif defined(__GNUC__)
48: #define HOST_OS "Linux"
49: #elif defined(_WIN32)
50: #define HOST_OS "Win32"
51: #else
52: #define HOST_OS "UNKNOWN"
53: #endif
54:
55: #define TRUE 1
56: #define FALSE 0
57:
58: #define M8(x) (x & 0xFF)
59:
60: /* various connection types */
61: #define XBEE_LOCAL_AT      0x88
62: #define XBEE_LOCAL_ATREQ  0x08
63: #define XBEE_LOCAL_ATQUE  0x09
64:
65: #define XBEE_REMOTE_AT     0x97
66: #define XBEE_REMOTE_ATREQ 0x17
67:
68: #define XBEE_MODEM_STATUS 0x8A
69:
70: #define XBEE_TX_STATUS     0x89
71: #define XBEE_64BIT_DATATX 0x00
72: #define XBEE_64BIT_DATARX 0x80
73: #define XBEE_16BIT_DATATX 0x01
74: #define XBEE_16BIT_DATARX 0x81
75: #define XBEE_64BIT_IO     0x82
76: #define XBEE_16BIT_IO     0x83
77:
78: typedef struct xbee_hnd* xbee_hnd;
79:
80: #define __LIBXBEE_API_H
81: #include "xbee.h"
82:
83: typedef struct t_threadList t_threadList;
84: struct t_threadList {
85:     xbee_thread_t thread;

```

```

86:     t_threadList *next;
87: };
88:
89: struct xbee_hnd {
90:     xbee_file_t tty;
91: #ifdef __GNUC__ /* ---- */
92:     int ttyfd;
93: #else /* ----- */
94:     int ttyr;
95:     int ttyw;
96:     int ttyeof;
97:
98:     OVERLAPPED ttyovrw;
99:     OVERLAPPED ttyovrr;
100:    OVERLAPPED ttyovrs;
101: #endif /* ----- */
102:
103:     char *path; /* serial port path */
104:
105:     xbee_mutex_t logmutex;
106:     FILE *log;
107:     int logfd;
108:
109:     xbee_mutex_t conmutex;
110:     xbee_con *conlist;
111:
112:     xbee_mutex_t pktmutex;
113:     xbee_pkt *pktlist;
114:     xbee_pkt *pktlast;
115:     int pktcount;
116:
117:     xbee_mutex_t sendmutex;
118:
119:     xbee_thread_t listent;
120:
121:     xbee_thread_t threadt;
122:     xbee_mutex_t threadmutex;
123:     xbee_sem_t threadsem;
124:     t_threadList *threadList;
125:
126:     int run;
127:
128:     int oldAPI;
129:     char cmdSeq;
130:     int cmdTime;
131:
132:     /* ready flag.
133:      needs to be set to -1 so that the listen thread can begin. */
134:     volatile int xbee_ready;
135:
136:     xbee_hnd next;
137: };
138: xbee_hnd default_xbee = NULL;
139: xbee_mutex_t xbee_hnd_mutex;
140:
141: typedef struct t_data t_data;
142: struct t_data {
143:     unsigned char data[128];
144:     unsigned int length;
145: };
146:
147: typedef struct t_LTinfo t_LTinfo;
148: struct t_LTinfo {
149:     int i;
150:     xbee_hnd xbee;
151: };
152:
153: typedef struct t_CBinfo t_CBinfo;
154: struct t_CBinfo {
155:     xbee_hnd xbee;
156:     xbee_con *con;
157: };
158:
159: typedef struct t_callback_list t_callback_list;
160: struct t_callback_list {
161:     xbee_pkt *pkt;
162:     t_callback_list *next;
163: };
164:
165: static void *Xmalloc2(xbee_hnd xbee, size_t size);
166: static void *Xcalloc2(xbee_hnd xbee, size_t size);
167: static void *Xrealloc2(xbee_hnd xbee, void *ptr, size_t size);
168: static void Xfree2(void **ptr);
169: #define Xmalloc(x)      Xmalloc2(xbee,(x))
170: #define Xcalloc(x)     Xcalloc2(xbee,(x))

```

```

171: #define Xrealloc(x,y) Xrealloc2(xbee,(x),(y))
172: #define Xfree(x) Xfree2((void *)&x)
173:
174: static void xbee_logf(xbee_hnd xbee, const char *logformat, int unlock, const char *file,
175: const int line, const char *function, char *format, ...);
176: #define LOG_FORMAT "[%s:%d] %s(): %s"
177: #define xbee_log(...) xbee_logf(xbee,LOG_FORMAT"\n",1,__FILE__,__LINE__,__FUNCTION__,__VA_ARGS__)
178: #define xbee_logc(...) xbee_logf(xbee,LOG_FORMAT,0,__FILE__,__LINE__,__FUNCTION__,__VA_ARGS__)
179: #define xbee_logcf(xbee) \
180: fprintf((xbee)->log,"\n"); \
181: xbee_mutex_unlock((xbee)->logmutex); \
182:
183: #define xbee_perror(str) \
184: if (xbee) xbee_log("%s:%s",str,strerror(errno)); \
185: perror(str); \
186:
187: static int xbee_startAPI(xbee_hnd xbee);
188:
189: static int xbee_sendAT(xbee_hnd xbee, char *command, char *retBuf, int retBuflen);
190: static int xbee_sendATdelay(xbee_hnd xbee, int guardTime, char *command, char *retBuf, int retBuflen);
191:
192: static int xbee_parse_io(xbee_hnd xbee, xbee_pkt *p, unsigned char *d,
193: int maskOffset, int sampleOffset, int sample);
194:
195: static void xbee_thread_watch(t_LTinfo *info);
196: static void xbee_listen_wrapper(t_LTinfo *info);
197: static int xbee_listen(xbee_hnd xbee, t_LTinfo *info);
198: static unsigned char xbee_getbyte(xbee_hnd xbee);
199: static unsigned char xbee_getrawbyte(xbee_hnd xbee);
200: static int xbee_matchpktcon(xbee_hnd xbee, xbee_pkt *pkt, xbee_con *con);
201:
202: static t_data *xbee_make_pkt(xbee_hnd xbee, unsigned char *data, int len);
203: static int xbee_send_pkt(xbee_hnd xbee, t_data *pkt, xbee_con *con);
204: static void xbee_callbackWrapper(t_CBinfo *info);
205:
206: /* these functions can be found in the xsys files */
207: static int init_serial(xbee_hnd xbee, int baudrate);
208: static int xbee_select(xbee_hnd xbee, struct timeval *timeout);
209:
210: #ifdef __GNUC__ /* ---- */
211: #include "xsys/linux.c"
212: #else /* ----- */
213: #include "xsys/win32.c"
214: #endif /* ----- */

```