```
/*
  libxbee - a C library to aid the use of Digi's Series 1 XBee modules
            running in API mode (AP=2).

  Copyright (C) 2009  Attie Grande (attie@attie.co.uk)

  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 3 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, see <http://www.gnu.org/licenses/>.
*/

#include <stdio.h>
#include <stdlib.h>

#include <stdarg.h>

#include <string.h>
#include <fcntl.h>
#include <errno.h>
#include <signal.h>

#ifdef __GNUC__ /* ---- */
#include <unistd.h>
#include <termios.h>
#define __USE_GNU
#include <pthread.h>
#undef __USE_GNU
#include <sys/time.h>
#else /* -------------- */
#include <Windows.h>
#include <io.h>
#include <time.h>
#include <sys/timeb.h>
#endif /* ------------- */

#ifdef __UMAKEFILE
  #define HOST_OS "Embedded"
#elif defined(__GNUC__)
  #define HOST_OS "Linux"
#elif defined(_WIN32)
  #define HOST_OS "Win32"
#else
  #define HOST_OS "UNKNOWN"
#endif

#define TRUE 1
#define FALSE 0

#define M8(x) (x & 0xFF)

/* various connection types */
#define XBEE_LOCAL_AT     0x88
#define XBEE_LOCAL_ATREQ  0x08
#define XBEE_LOCAL_ATQUE  0x09

#define XBEE_REMOTE_AT    0x97
#define XBEE_REMOTE_ATREQ 0x17

#define XBEE_MODEM_STATUS 0x8A

#define XBEE_TX_STATUS    0x89
#define XBEE_64BIT_DATATX 0x00
#define XBEE_64BIT_DATARX 0x80
#define XBEE_16BIT_DATATX 0x01
#define XBEE_16BIT_DATARX 0x81
#define XBEE_64BIT_IO     0x82
#define XBEE_16BIT_IO     0x83

typedef struct xbee_hnd* xbee_hnd;

#define __LIBXBEE_API_H
#include "xbee.h"

typedef struct t_threadList t_threadList;
struct t_threadList {
  xbee_thread_t thread;
```

```
  t_threadList *next;
};

struct xbee_hnd {
  xbee_file_t tty;
#ifdef __GNUC__ /* ---- */
  int ttyfd;
#else /* -------------- */
  int ttyr;
  int ttyw;
  int ttyeof;

  OVERLAPPED ttyovrw;
  OVERLAPPED ttyovrr;
  OVERLAPPED ttyovrs;
#endif /* ------------- */

  char *path; /* serial port path */

  xbee_mutex_t logmutex;
  FILE *log;
  int logfd;

  xbee_mutex_t conmutex;
  xbee_con *conlist;

  xbee_mutex_t pktmutex;
  xbee_pkt *pktlist;
  xbee_pkt *pktlast;
  int pktcount;

  xbee_mutex_t sendmutex;

  xbee_thread_t listent;

  xbee_thread_t threadt;
  xbee_mutex_t  threadmutex;
  xbee_sem_t    threadsem;
  t_threadList *threadList;

  int run;

  int oldAPI;
  char cmdSeq;
  int cmdTime;

  /* ready flag.
     needs to be set to -1 so that the listen thread can begin. */
  volatile int xbee_ready;

  xbee_hnd next;
};
xbee_hnd default_xbee = NULL;
xbee_mutex_t xbee_hnd_mutex;

typedef struct t_data t_data;
struct t_data {
  unsigned char data[128];
  unsigned int length;
};

typedef struct t_LTinfo t_LTinfo;
struct t_LTinfo {
  int i;
  xbee_hnd xbee;
};

typedef struct t_CBinfo t_CBinfo;
struct t_CBinfo {
  xbee_hnd xbee;
  xbee_con *con;
};

typedef struct t_callback_list t_callback_list;
struct t_callback_list {
  xbee_pkt *pkt;
  t_callback_list *next;
};

static void *Xmalloc2(xbee_hnd xbee, size_t size);
static void *Xcalloc2(xbee_hnd xbee, size_t size);
static void *Xrealloc2(xbee_hnd xbee, void *ptr, size_t size);
static void Xfree2(void **ptr);
#define Xmalloc(x)     Xmalloc2(xbee,(x))
#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 /* ------------- */
```