```c
/*
  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>
#include <pthread.h>
#include <sys/time.h>
#else /* -------------- */
#include <Windows.h>
#include <io.h>
#include <time.h>
#include <sys/timeb.h>
#endif /* ------------- */

#include "xbee.h"

#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 ISREADY                                          \
  if (!xbee_ready) {                                     \
    if (stderr) fprintf(stderr,"libxbee: Run xbee_setup() first!...\n"); \
    exit(1);                                             \
  }

#define M8(x) (x & 0xFF)
#define FDO(x,y,z)                                     \
  if (((x) = fdopen((y),(z))) == NULL) {               \
    perror("fopen()");                                 \
    return(-1);                                        \
  }
#define FO(x,y,z)                                      \
  if (((x) = open((y),(z))) == -1) {                   \
    perror("open()");                                  \
    return(-1);                                        \
  }

/* 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

```

```c
#define XBEE_TX_STATUS    0x89
#define XBEE_64BIT_DATATX 0x00
#define XBEE_64BIT_DATA   0x80
#define XBEE_16BIT_DATATX 0x01
#define XBEE_16BIT_DATA   0x81

#define XBEE_64BIT_IO     0x82
#define XBEE_16BIT_IO     0x83

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

typedef struct t_info t_info;
struct t_info {
  int i;
};

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

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

  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;
  int listenrun;

  int oldAPI;
  char cmdSeq;
  int cmdTime;
} xbee;

/* ready flag.
   needs to be set to -1 so that the listen thread can begin.
   then 1 so that functions can be used (after setup of course...) */
volatile int xbee_ready = 0;

static void *Xmalloc(size_t size);
static void *Xcalloc(size_t size);
static void *Xrealloc(void *ptr, size_t size);
static void Xfree2(void **ptr);
#define Xfree(x) Xfree2((void **)&x)

static void xbee_logf(const char *logformat, int unlock, const char *file,
                      const int line, const char *function, char *format, ...);
#define xbee_log(...) xbee_logf("[%s:%d] %s(): %s\n",1,__FILE__,__LINE__,__FUNCTION__,__VA_ARGS__)
#define xbee_logc(...) xbee_logf("[%s:%d] %s(): %s",0,__FILE__,__LINE__,__FUNCTION__,__VA_ARGS__)
#define xbee_logcf()                     \
  fprintf(xbee.log,"\n");                \
  xbee_mutex_unlock(xbee.logmutex);  \

static int xbee_startAPI(void);

static int xbee_sendAT(char *command, char *retBuf, int retBuflen);
static int xbee_sendATdelay(int guardTime, char *command, char *retBuf, int retBuflen);

static int xbee_parse_io(xbee_pkt *p, unsigned char *d, int maskOffset, int sampleOffset, int sample);
static void xbee_listen_wrapper(t_info *info);
static int xbee_listen(t_info *info);
static unsigned char xbee_getbyte(void);
static unsigned char xbee_getrawbyte(void);
static int xbee_matchpktcon(xbee_pkt *pkt, xbee_con *con);

static t_data *xbee_make_pkt(unsigned char *data, int len);
static int xbee_send_pkt(t_data *pkt, xbee_con *con);
static void xbee_callbackWrapper(xbee_con *con);

/* these functions can be found in the xsys files */
static int init_serial(int baudrate);
static int xbee_select(struct timeval *timeout);

#ifdef __GNUC__ /* ---- */
#include "xsys/linux.c"
#else /* -------------- */
#include "xsys\win32.c"
#endif /* ------------- */
```