```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/>.
*/

/*  this file contains code that is used by Win32 ONLY */
#ifndef _WIN32
#error "This file should only be used on a Win32 system"
#endif

/* ################################################################# */
/* ### Win32 Code ################################################## */
/* ################################################################# */

#pragma comment(lib, "Advapi32.lib")
#pragma comment(lib, "User32.lib")

#define dllid   "attie-co-uk.libxbee"
#define dlldesc "libxbee - XBee API Library"
/* libxbee’s GUID is {7A6E25AA-ECB5-4370-87B5-A1D31840FE23} */
#define dllGUID "{7A6E25AA-ECB5-4370-87B5-A1D31840FE23}"

HMODULE glob_hModule = NULL;

/* this uses miliseconds not microseconds... */
#define usleep(a)                 Sleep((a)/1000)

#define xbee_thread_create(a,b,c) (((a) = CreateThread(NULL,0,(void *)(b),(void *)(c),0,NULL)) == NULL)
#define xbee_thread_cancel(a,b)   TerminateThread((a),(b))
#define xbee_thread_join(a)       WaitForSingleObject((a),INFINITE)
#define xbee_thread_tryjoin(a)    WaitForSingleObject((a),0)

#define xbee_mutex_init(a)        (!InitializeCriticalSectionAndSpinCount(&(a),0))
#define xbee_mutex_destroy(a)     DeleteCriticalSection(&(a))
#define xbee_mutex_lock(a)        EnterCriticalSection(&(a))
#define xbee_mutex_trylock(a)     (!TryEnterCriticalSection(&(a)))
#define xbee_mutex_unlock(a)      LeaveCriticalSection(&(a))

#define xbee_sem_init(a)          (((a) = CreateEvent(NULL,FALSE,FALSE,NULL)) == NULL)
#define xbee_sem_destroy(a)       CloseHandle((a))
#define xbee_sem_wait(a)          WaitForSingleObject((a),INFINITE)
#define xbee_sem_wait1sec(a)      WaitForSingleObject((a),1000)
#define xbee_sem_post(a)          SetEvent((a))

#define xbee_cond_init(a)         InitializeConditionVariable(&(a))
#define xbee_cond_destroy(a)
#define xbee_cond_wait(a,b)       SleepConditionVariableCS(&(a),&(b),INFINITE)
#define xbee_cond_signal(a)       WakeConditionVariable(&(a))
#define xbee_cond_broadcast(a)    WakeAllConditionVariable(&(a))

#define xbee_feof(a)              (xbee->ttyeof)
#define xbee_ferror(a)            (0)
#define xbee_close(a)             CloseHandle((a))

typedef struct win32_callback_info win32_callback_info;
struct win32_callback_info {
  xbee_con *con;
  HWND hWnd;
  UINT uMsg;
  win32_callback_info *next;
};

win32_callback_info *callbackMap = NULL;
xbee_mutex_t callbackmutex;
char callbackmutexInitialized = 0;
```