

```

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: /* #####
22: /* ### Win32 DLL Code #####
23: /* #####
24:
25: /* this file contains code that is used by Win32 ONLY */
26: #ifndef _WIN32
27: #error "This file should only be used on a Win32 system"
28: #endif
29:
30: int ver(HWND hwnd, HINSTANCE hinst, LPWSTR lpszCmdLine, int nCmdShow) {
31:     char t[256];
32:     sprintf(t, "libxbee.dll\n%s\n%s", xbee_svn_version(), xbee_build_info());
33:     MessageBox(NULL, t, "libxbee Win32 DLL", MB_OK);
34:     return 0;
35: }
36:
37: /* this gets called when the dll is loaded... */
38: BOOL APIENTRY DllMain(HANDLE hModule, DWORD dwReason, LPVOID lpReserved) {
39:     if ((dwReason == DLL_PROCESS_DETACH) && xbee_ready == 1) {
40:         /* ensure that libxbee has been shut down nicely */
41:         xbee_end();
42:     } else if (glob_hModule == NULL && (dwReason == DLL_PROCESS_ATTACH || dwReason == DLL_THREAD_ATTACH)) {
43:         /* keep a handle on the module */
44:         glob_hModule = (HMODULE)hModule;
45:         xbee_mutex_init(callbackmutex);
46:     }
47:     return TRUE;
48: }
49:
50: HRESULT DllCanUnloadNow(void) {
51:     return !xbee_ready;
52: }
53:
54: /* #####
55: /* ### Win32 DLL COM Code #####
56: /* #####
57:
58: /* this function is from this tutorial:
59:     http://www.codeguru.com/Cpp/COM-Tech/activex/tutorials/article.php/c5567 */
60: BOOL RegWriteKey(HKEY roothk, const char *lpSubKey, LPCTSTR val_name,
61:                 DWORD dwType, void *lpvData, DWORD dwDataSize) {
62:     /* roothk:     HKEY_CLASSES_ROOT, HKEY_LOCAL_MACHINE, etc
63:        lpSubKey:  the key relative to 'roothk'
64:        val_name:  the key value name where the data will be written
65:        dwType:    REG_SZ, REG_BINARY, etc.
66:        lpvData:   a pointer to the data buffer
67:        dwDataSize: the size of the data pointed to by lpvData */
68:     HKEY hk;
69:     if (ERROR_SUCCESS != RegCreateKey(roothk, lpSubKey, &hk) ) return FALSE;
70:     if (ERROR_SUCCESS != RegSetValueEx(hk, val_name, 0, dwType, (CONST BYTE *)lpvData, dwDataSize) ) return FALSE;
71:     if (ERROR_SUCCESS != RegCloseKey(hk) ) return FALSE;
72:     return TRUE;
73: }
74:
75: /* this is used by the regsrv32 application */
76: STDAPI DllRegisterServer(void) {
77:     char key[MAX_PATH];
78:     char value[MAX_PATH];
79:
80:     wsprintf(key, "CLSID\\%s", dllGUID);
81:     wsprintf(value, "%s", dlldesc);
82:     RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));
83:
84:     wsprintf(key, "CLSID\\%s\\InprocServer32", dllGUID);
85:     GetModuleFileName(glob_hModule, value, MAX_PATH);

```

```
86: RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));
87:
88: wsprintf(key, "CLSID\\%s\\ProgId", dllGUID);
89: lstrcpy(value, dllid);
90: RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));
91:
92: lstrcpy(key, dllid);
93: lstrcpy(value, dllid);
94: RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)value, lstrlen(value));
95:
96: wsprintf(key, "%s\\CLSID", dllid);
97: RegWriteKey(HKEY_CLASSES_ROOT, key, NULL, REG_SZ, (void *)dllGUID, lstrlen(dllGUID));
98:
99: return S_OK;
100: }
101:
102: /* this is used by the regsrv32 application */
103: STDAPI DllUnregisterServer(void) {
104:     char key[MAX_PATH];
105:     char value[MAX_PATH];
106:
107:     wsprintf(key, "%s\\CLSID", dllid);
108:     RegDeleteKey(HKEY_CLASSES_ROOT, key);
109:
110:     wsprintf(key, "%s", dllid);
111:     RegDeleteKey(HKEY_CLASSES_ROOT, key);
112:
113:     wsprintf(key, "CLSID\\%s\\InprocServer32", dllGUID);
114:     RegDeleteKey(HKEY_CLASSES_ROOT, key);
115:
116:     wsprintf(key, "CLSID\\%s\\ProgId", dllGUID);
117:     RegDeleteKey(HKEY_CLASSES_ROOT, key);
118:
119:     wsprintf(key, "CLSID\\%s", dllGUID);
120:     RegDeleteKey(HKEY_CLASSES_ROOT, key);
121:
122:     return S_OK;
123: }
```