

```
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: #include <string.h>
24:
25: #include "xbee.h"
26:
27: int main(int argc, char *argv[]) {
28:     xbee_con *con, *con2;
29:     xbee_pkt *pkt, *p;
30:
31:     if (xbee_setuplog("/dev/ttyUSB0",57600,2) == -1) {
32:         perror("xbee_setuplog()");
33:         exit(1);
34:     }
35:     if (argc >= 2 && !strcmp(argv[1],"sleep")) {
36:         for (;;) {
37:             sleep(86400); /* sleep for a day... forever :) */
38:         }
39:     }
40:
41:     /*if ((con = xbee_newcon(NULL,'X',xbee_localAT)) == (void *)-1) {
42:         printf("error creating connection...\n");
43:         exit(1);
44:     }
45:
46:     while(1){sleep(10);}
47:
48:     xbee_senddata(con,"CH%c",0x0C);
49:     sleep(1);
50:     xbee_senddata(con,"ID%c%c",0x33, 0x32);
51:     sleep(1);
52:     xbee_senddata(con,"DH%c%c%c%c",0x00,0x00,0x00,0x00);
53:     sleep(1);
54:     xbee_senddata(con,"DL%c%c%c%c",0x00,0x00,0x00,0x00);
55:     sleep(1);
56:     xbee_senddata(con,"MY%c%c",0x00,0x00);
57:     sleep(1);
58:     // SH - read only
59:     // SL - read only
60:     xbee_senddata(con,"RR%c",0x00);
61:     sleep(1);
62:     xbee_senddata(con,"RN%c",0x00);
63:     sleep(1);
64:     xbee_senddata(con,"MM%c",0x00);
65:     sleep(1);
66:     xbee_senddata(con,"NT%c",0x19);
67:     sleep(1);
68:     xbee_senddata(con,"NO%c",0x00);
69:     sleep(1);
70:     xbee_senddata(con,"CE%c",0x00);
71:     sleep(1);
72:     xbee_senddata(con,"SC%c%c",0x1F,0xFE);
73:     sleep(1);
74:     xbee_senddata(con,"SD%c",0x04);
75:     sleep(1);
76:     xbee_senddata(con,"A1%c",0x00);
77:     sleep(1);
78:     xbee_senddata(con,"A2%c",0x00);
79:     sleep(1);
80:     // AI - read only
81:     xbee_senddata(con,"EE%c",0x00);
82:     sleep(1);
83:     //xbee_senddata(con,"KY%c",0x00);
84:     //sleep(1);
85:     xbee_senddata(con,"NI%s","TIGGER");
```

```
86:  sleep(1);
87:  xbee_senddata(con,"PL%c",0x04);
88:  sleep(1);
89:  xbee_senddata(con,"CA%c",0x2C);
90:  sleep(1);
91:  xbee_senddata(con,"SM%c",0x00);
92:  sleep(1);
93:  xbee_senddata(con,"ST%c%c",0x13,0x88);
94:  sleep(1);
95:  xbee_senddata(con,"SP%c%c",0x00,0x00);
96:  sleep(1);
97:  xbee_senddata(con,"DP%c%c",0x03,0xE8);
98:  sleep(1);
99:  xbee_senddata(con,"SO%c",0x00);
100: sleep(1);
101: xbee_senddata(con,"BD%c",0x06);
102: sleep(1);
103: xbee_senddata(con,"RO%c",0x03);
104: sleep(1);
105: xbee_senddata(con,"AP%c",0x02);
106: sleep(1);
107: xbee_senddata(con,"PR%c",0xFF);
108: sleep(1);
109: xbee_senddata(con,"D8%c",0x00);
110: sleep(1);
111: xbee_senddata(con,"D7%c",0x01);
112: sleep(1);
113: xbee_senddata(con,"D6%c",0x00);
114: sleep(1);
115: xbee_senddata(con,"D5%c",0x01);
116: sleep(1);
117: xbee_senddata(con,"D4%c",0x00);
118: sleep(1);
119: xbee_senddata(con,"D3%c",0x00);
120: sleep(1);
121: xbee_senddata(con,"D2%c",0x00);
122: sleep(1);
123: xbee_senddata(con,"D1%c",0x00);
124: sleep(1);
125: xbee_senddata(con,"D0%c",0x00);
126: sleep(1);
127: xbee_senddata(con,"IU%c",0x00);
128: sleep(1);
129: xbee_senddata(con,"IT%c",0x01);
130: sleep(1);
131: xbee_senddata(con,"IC%c",0x00);
132: sleep(1);
133: xbee_senddata(con,"IR%c%c",0x00,0x00);
134: sleep(1);
135: xbee_senddata(con,"IA%c%c%c%c%c%c%c%c",0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF);
136: sleep(1);
137: xbee_senddata(con,"T0%c",0xFF);
138: sleep(1);
139: xbee_senddata(con,"T1%c",0xFF);
140: sleep(1);
141: xbee_senddata(con,"T2%c",0xFF);
142: sleep(1);
143: xbee_senddata(con,"T3%c",0xFF);
144: sleep(1);
145: xbee_senddata(con,"T4%c",0xFF);
146: sleep(1);
147: xbee_senddata(con,"T5%c",0xFF);
148: sleep(1);
149: xbee_senddata(con,"T6%c",0xFF);
150: sleep(1);
151: xbee_senddata(con,"T7%c",0xFF);
152: sleep(1);
153: xbee_senddata(con,"P0%c",0x01);
154: sleep(1);
155: xbee_senddata(con,"P1%c",0x00);
156: sleep(1);
157: xbee_senddata(con,"PT%c",0xFF);
158: sleep(1);
159: xbee_senddata(con,"RP%c",0x28);
160: sleep(1);
161: // VR - read only
162: // HV - read only
163: // DB - read only
164: // EC - read only
165: // EA - read only
166: // DD - read only
167: xbee_senddata(con,"CT%c",0x64);
168: sleep(1);
169: xbee_senddata(con,"GT%c%c",0x03,0xE8);
170: sleep(1);
```

```
171: xbee_senddata(con,"CC%c",0x2B);
172: sleep(1);
173:
174: sleep(10);
175: */
176:
177: /* test 64bit IO and Data */
178: con = xbee_newcon('I',xbee_64bitIO, 0x0013A200, 0x403af247);
179: con2 = xbee_newcon('I',xbee_64bitData, 0x0013A200, 0x403af247);
180:
181: while (1) {
182:     while ((pkt = xbee_getpacket(con)) != NULL) {
183:         int i;
184:         for (i = 0; i < pkt->samples; i++) {
185:             int m;
186:             for (m = 0; m <= 8; m++) {
187:                 if (xbee_hasdigital(pkt,i,m)) printf("D%d: %d ",m,xbee_getdigital(pkt,i,m));
188:             }
189: #define Vref 3.23
190:             for (m = 0; m <= 5; m++) {
191:                 if (xbee_hasanalog(pkt,i,m)) printf("A%d: %.2fv ",m,xbee_getanalog(pkt,i,m,Vref));
192:             }
193:             printf("\n");
194:         }
195:         if (xbee_senddata(con2, "the time is %d\r", time(NULL))) {
196:             printf("Error: xbee_senddata\n");
197:             return 1;
198:         }
199:         free(pkt);
200:         if (p) {
201:             switch (p->status) {
202:                 case 0x01: printf("XBee: txStatus: No ACK\n"); break;
203:                 case 0x02: printf("XBee: txStatus: CCA Failure\n"); break;
204:                 case 0x03: printf("XBee: txStatus: Purged\n"); break;
205:             }
206:             free(p);
207:         }
208:     }
209:     while ((pkt = xbee_getpacket(con2)) != NULL) {
210:         printf("he said '%s'\n", pkt->data);
211:         if (xbee_senddata(con2, "you said '%s'\r", pkt->data)) {
212:             printf("Error: xbee_senddata\n");
213:             return 1;
214:         }
215:         free(pkt);
216:         if (p) {
217:             switch (p->status) {
218:                 case 0x01: printf("XBee: txStatus: No ACK\n"); break;
219:                 case 0x02: printf("XBee: txStatus: CCA Failure\n"); break;
220:                 case 0x03: printf("XBee: txStatus: Purged\n"); break;
221:             }
222:             free(p);
223:         }
224:     }
225:     usleep(100);
226: }
227:
228: return 0;
229: }
```