

```
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 "globals.h"
22:
23: int main(int argc, char *argv[]) {
24:     xbee_con *con, *con2;
25:     xbee_pkt *pkt, *p;
26:
27:     if (xbee_setuplog("/dev/ttyUSB0",57600,2) == -1) {
28:         perror("xbee_setuplog()");
29:         exit(1);
30:     }
31:     if (argc >= 2 && !strcmp(argv[1],"sleep")) {
32:         for (;;) {
33:             sleep(86400); /* sleep for a day... forever :) */
34:         }
35:     }
36:
37:     /*if ((con = xbee_newcon(NULL,'X',xbee_localAT)) == (void *)-1) {
38:         printf("error creating connection...\n");
39:         exit(1);
40:     }
41:
42:     while(1){sleep(10);}
43:
44:     xbee_senddata(con,"CH%c",0x0C);
45:     sleep(1);
46:     xbee_senddata(con,"ID%c%c",0x33, 0x32);
47:     sleep(1);
48:     xbee_senddata(con,"DH%c%c%c%c",0x00,0x00,0x00,0x00);
49:     sleep(1);
50:     xbee_senddata(con,"DL%c%c%c%c",0x00,0x00,0x00,0x00);
51:     sleep(1);
52:     xbee_senddata(con,"MY%c%c",0x00,0x00);
53:     sleep(1);
54:     // SH - read only
55:     // SL - read only
56:     xbee_senddata(con,"RR%c",0x00);
57:     sleep(1);
58:     xbee_senddata(con,"RN%c",0x00);
59:     sleep(1);
60:     xbee_senddata(con,"MM%c",0x00);
61:     sleep(1);
62:     xbee_senddata(con,"NT%c",0x19);
63:     sleep(1);
64:     xbee_senddata(con,"NO%c",0x00);
65:     sleep(1);
66:     xbee_senddata(con,"CE%c",0x00);
67:     sleep(1);
68:     xbee_senddata(con,"SC%c%c",0x1F,0xFE);
69:     sleep(1);
70:     xbee_senddata(con,"SD%c",0x04);
71:     sleep(1);
72:     xbee_senddata(con,"A1%c",0x00);
73:     sleep(1);
74:     xbee_senddata(con,"A2%c",0x00);
75:     sleep(1);
76:     // AI - read only
77:     xbee_senddata(con,"EE%c",0x00);
78:     sleep(1);
79:     //xbee_senddata(con,"KY%c",0x00);
80:     //sleep(1);
81:     xbee_senddata(con,"NIS","TIGGER");
82:     sleep(1);
83:     xbee_senddata(con,"PL%c",0x04);
84:     sleep(1);
85:     xbee_senddata(con,"CA%c",0x2C);
```

```
86:  sleep(1);
87:  xbee_senddata(con,"SM%c",0x00);
88:  sleep(1);
89:  xbee_senddata(con,"ST%c%c",0x13,0x88);
90:  sleep(1);
91:  xbee_senddata(con,"SP%c%c",0x00,0x00);
92:  sleep(1);
93:  xbee_senddata(con,"DP%c%c",0x03,0xE8);
94:  sleep(1);
95:  xbee_senddata(con,"SO%c",0x00);
96:  sleep(1);
97:  xbee_senddata(con,"BD%c",0x06);
98:  sleep(1);
99:  xbee_senddata(con,"RO%c",0x03);
100: sleep(1);
101: xbee_senddata(con,"AP%c",0x02);
102: sleep(1);
103: xbee_senddata(con,"PR%c",0xFF);
104: sleep(1);
105: xbee_senddata(con,"D8%c",0x00);
106: sleep(1);
107: xbee_senddata(con,"D7%c",0x01);
108: sleep(1);
109: xbee_senddata(con,"D6%c",0x00);
110: sleep(1);
111: xbee_senddata(con,"D5%c",0x01);
112: sleep(1);
113: xbee_senddata(con,"D4%c",0x00);
114: sleep(1);
115: xbee_senddata(con,"D3%c",0x00);
116: sleep(1);
117: xbee_senddata(con,"D2%c",0x00);
118: sleep(1);
119: xbee_senddata(con,"D1%c",0x00);
120: sleep(1);
121: xbee_senddata(con,"D0%c",0x00);
122: sleep(1);
123: xbee_senddata(con,"IU%c",0x00);
124: sleep(1);
125: xbee_senddata(con,"IT%c",0x01);
126: sleep(1);
127: xbee_senddata(con,"IC%c",0x00);
128: sleep(1);
129: xbee_senddata(con,"IR%c%c",0x00,0x00);
130: sleep(1);
131: xbee_senddata(con,"IA%c%c%c%c%c%c%c%c",0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF);
132: sleep(1);
133: xbee_senddata(con,"T0%c",0xFF);
134: sleep(1);
135: xbee_senddata(con,"T1%c",0xFF);
136: sleep(1);
137: xbee_senddata(con,"T2%c",0xFF);
138: sleep(1);
139: xbee_senddata(con,"T3%c",0xFF);
140: sleep(1);
141: xbee_senddata(con,"T4%c",0xFF);
142: sleep(1);
143: xbee_senddata(con,"T5%c",0xFF);
144: sleep(1);
145: xbee_senddata(con,"T6%c",0xFF);
146: sleep(1);
147: xbee_senddata(con,"T7%c",0xFF);
148: sleep(1);
149: xbee_senddata(con,"P0%c",0x01);
150: sleep(1);
151: xbee_senddata(con,"P1%c",0x00);
152: sleep(1);
153: xbee_senddata(con,"PT%c",0xFF);
154: sleep(1);
155: xbee_senddata(con,"RP%c",0x28);
156: sleep(1);
157: // VR - read only
158: // HV - read only
159: // DB - read only
160: // EC - read only
161: // EA - read only
162: // DD - read only
163: xbee_senddata(con,"CT%c",0x64);
164: sleep(1);
165: xbee_senddata(con,"GT%c%c",0x03,0xE8);
166: sleep(1);
167: xbee_senddata(con,"CC%c",0x2B);
168: sleep(1);
169:
170: sleep(10);
```

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