

```
1: #include "globals.h"
2:
3: int main(int argc, char *argv[]) {
4:     xbee_con *con, *con2;
5:     xbee_pkt *pkt, *p;
6:
7:     if (xbee_setup("/dev/ttyUSB1",57600) == -1) {
8:         perror("xbee_setup()");
9:         exit(1);
10:    }
11:
12:    /*if ((con = xbee_newcon(NULL,'X',xbee_localAT)) == (void *)-1) {
13:        printf("error creating connection...\n");
14:        exit(1);
15:    }
16:
17:    while(1){sleep(10);}
18:
19:    xbee_senddata(con,"CH%c",0x0C);
20:    sleep(1);
21:    xbee_senddata(con,"ID%c%c",0x33, 0x32);
22:    sleep(1);
23:    xbee_senddata(con,"DH%c%c%c%c",0x00,0x00,0x00,0x00);
24:    sleep(1);
25:    xbee_senddata(con,"DL%c%c%c%c",0x00,0x00,0x00,0x00);
26:    sleep(1);
27:    xbee_senddata(con,"MY%c%c",0x00,0x00);
28:    sleep(1);
29:    // SH - read only
30:    // SL - read only
31:    xbee_senddata(con,"RR%c",0x00);
32:    sleep(1);
33:    xbee_senddata(con,"RN%c",0x00);
34:    sleep(1);
35:    xbee_senddata(con,"MM%c",0x00);
36:    sleep(1);
37:    xbee_senddata(con,"NT%c",0x19);
38:    sleep(1);
39:    xbee_senddata(con,"NO%c",0x00);
40:    sleep(1);
41:    xbee_senddata(con,"CE%c",0x00);
42:    sleep(1);
43:    xbee_senddata(con,"SC%c%c",0x1F,0xFE);
44:    sleep(1);
45:    xbee_senddata(con,"SD%c",0x04);
46:    sleep(1);
47:    xbee_senddata(con,"A1%c",0x00);
48:    sleep(1);
49:    xbee_senddata(con,"A2%c",0x00);
50:    sleep(1);
51:    // AI - read only
52:    xbee_senddata(con,"EE%c",0x00);
53:    sleep(1);
54:    //xbee_senddata(con,"KY%c",0x00);
55:    //sleep(1);
56:    xbee_senddata(con,"NIS", "TIGGER");
57:    sleep(1);
58:    xbee_senddata(con,"PL%c",0x04);
59:    sleep(1);
60:    xbee_senddata(con,"CA%c",0x2C);
61:    sleep(1);
62:    xbee_senddata(con,"SM%c",0x00);
63:    sleep(1);
64:    xbee_senddata(con,"ST%c%c",0x13,0x88);
65:    sleep(1);
66:    xbee_senddata(con,"SP%c%c",0x00,0x00);
67:    sleep(1);
68:    xbee_senddata(con,"DP%c%c",0x03,0xE8);
69:    sleep(1);
70:    xbee_senddata(con,"SO%c",0x00);
71:    sleep(1);
72:    xbee_senddata(con,"BD%c",0x06);
73:    sleep(1);
74:    xbee_senddata(con,"RO%c",0x03);
75:    sleep(1);
76:    xbee_senddata(con,"AP%c",0x02);
77:    sleep(1);
78:    xbee_senddata(con,"PR%c",0xFF);
79:    sleep(1);
80:    xbee_senddata(con,"D8%c",0x00);
81:    sleep(1);
82:    xbee_senddata(con,"D7%c",0x01);
83:    sleep(1);
84:    xbee_senddata(con,"D6%c",0x00);
85:    sleep(1);
```

```
86: xbee_senddata(con,"D5%c",0x01);
87: sleep(1);
88: xbee_senddata(con,"D4%c",0x00);
89: sleep(1);
90: xbee_senddata(con,"D3%c",0x00);
91: sleep(1);
92: xbee_senddata(con,"D2%c",0x00);
93: sleep(1);
94: xbee_senddata(con,"D1%c",0x00);
95: sleep(1);
96: xbee_senddata(con,"D0%c",0x00);
97: sleep(1);
98: xbee_senddata(con,"IU%c",0x00);
99: sleep(1);
100: xbee_senddata(con,"IT%c",0x01);
101: sleep(1);
102: xbee_senddata(con,"IC%c",0x00);
103: sleep(1);
104: xbee_senddata(con,"IR%c%c",0x00,0x00);
105: sleep(1);
106: xbee_senddata(con,"IA%c%c%c%c%c%c%c",0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF);
107: sleep(1);
108: xbee_senddata(con,"T0%c",0xFF);
109: sleep(1);
110: xbee_senddata(con,"T1%c",0xFF);
111: sleep(1);
112: xbee_senddata(con,"T2%c",0xFF);
113: sleep(1);
114: xbee_senddata(con,"T3%c",0xFF);
115: sleep(1);
116: xbee_senddata(con,"T4%c",0xFF);
117: sleep(1);
118: xbee_senddata(con,"T5%c",0xFF);
119: sleep(1);
120: xbee_senddata(con,"T6%c",0xFF);
121: sleep(1);
122: xbee_senddata(con,"T7%c",0xFF);
123: sleep(1);
124: xbee_senddata(con,"P0%c",0x01);
125: sleep(1);
126: xbee_senddata(con,"P1%c",0x00);
127: sleep(1);
128: xbee_senddata(con,"PT%c",0xFF);
129: sleep(1);
130: xbee_senddata(con,"RP%c",0x28);
131: sleep(1);
132: // VR - read only
133: // HV - read only
134: // DB - read only
135: // EC - read only
136: // EA - read only
137: // DD - read only
138: xbee_senddata(con,"CT%c",0x64);
139: sleep(1);
140: xbee_senddata(con,"GT%c%c",0x03,0xE8);
141: sleep(1);
142: xbee_senddata(con,"CC%c",0x2B);
143: sleep(1);
144:
145: sleep(10);
146: */
147:
148: /* test local AT */
149: con = xbee_newcon('I',xbee_localAT);
150: p = xbee_senddata(con,"NI");
151: if (p && p->status != 0) {
152:     printf("local AT error (0x%02X)\n", p->status);
153: } else if (p) {
154:     printf("local node identifier: %s\n",p->data);
155:     free(p);
156: }
157:
158: /* test remote AT */
159: con = xbee_newcon('I',xbee_remoteAT, 0x0013A200, 0x40081826);
160: p = xbee_senddata(con,"NI");
161: if (p && p->status != 0) {
162:     printf("remote AT error (0x%02X)\n", p->status);
163: } else if (p) {
164:     printf("remote node identifier: %s\n",p->data);
165:     free(p);
166: }
167:
168: /* test 64bit IO and Data */
169: con = xbee_newcon('I',xbee_64bitIO, 0x0013A200, 0x40081826);
170: con2 = xbee_newcon('I',xbee_64bitData, 0x0013A200, 0x40081826);
```

```

171:
172: while (1) {
173:     while ((pkt = xbee_getpacket(con)) != NULL) {
174:         printf("----- got one!... CON -----\\n");
175:         if (pkt->Iomask & 0x0001) printf("Digital 0: %c\\n", ((pkt->IOdata & 0x0001)?'1':'0'));
176:         if (pkt->Iomask & 0x0002) printf("Digital 1: %c\\n", ((pkt->IOdata & 0x0002)?'1':'0'));
177:         if (pkt->Iomask & 0x0004) printf("Digital 2: %c\\n", ((pkt->IOdata & 0x0004)?'1':'0'));
178:         if (pkt->Iomask & 0x0008) printf("Digital 3: %c\\n", ((pkt->IOdata & 0x0008)?'1':'0'));
179:         if (pkt->Iomask & 0x0010) printf("Digital 4: %c\\n", ((pkt->IOdata & 0x0010)?'1':'0'));
180:         if (pkt->Iomask & 0x0020) printf("Digital 5: %c\\n", ((pkt->IOdata & 0x0020)?'1':'0'));
181:         if (pkt->Iomask & 0x0040) printf("Digital 6: %c\\n", ((pkt->IOdata & 0x0040)?'1':'0'));
182:         if (pkt->Iomask & 0x0080) printf("Digital 7: %c\\n", ((pkt->IOdata & 0x0080)?'1':'0'));
183:         if (pkt->Iomask & 0x0100) printf("Digital 8: %c\\n", ((pkt->IOdata & 0x0100)?'1':'0'));
184:         if (pkt->Iomask & 0x0200) printf("Analog 0: %.2fv\\n", (3.3/1023)*pkt->IOanalog[0]);
185:         if (pkt->Iomask & 0x0400) printf("Analog 1: %.2fv\\n", (3.3/1023)*pkt->IOanalog[1]);
186:         if (pkt->Iomask & 0x0800) printf("Analog 2: %.2fv\\n", (3.3/1023)*pkt->IOanalog[2]);
187:         if (pkt->Iomask & 0x1000) printf("Analog 3: %.2fv\\n", (3.3/1023)*pkt->IOanalog[3]);
188:         if (pkt->Iomask & 0x2000) printf("Analog 4: %.2fv\\n", (3.3/1023)*pkt->IOanalog[4]);
189:         if (pkt->Iomask & 0x4000) printf("Analog 5: %.2fv\\n", (3.3/1023)*pkt->IOanalog[5]);
190:         p = xbee_senddata(con2, "the time is %d\\r", time(NULL));
191:         free(pkt);
192:         if (p) {
193:             switch (p->status) {
194:                 case 0x00: printf("XBee: txStatus: Success!\\n"); break;
195:                 case 0x01: printf("XBee: txStatus: No ACK\\n"); break;
196:                 case 0x02: printf("XBee: txStatus: CCA Failure\\n"); break;
197:                 case 0x03: printf("XBee: txStatus: Purged\\n"); break;
198:             }
199:             free(p);
200:         }
201:     }
202:     while ((pkt = xbee_getpacket(con2)) != NULL) {
203:         printf("----- got one!... CON2 -----\\n");
204:         printf("he said '%s'\\n", pkt->data);
205:         p = xbee_senddata(con2, "you said '%s'\\r", pkt->data);
206:         free(pkt);
207:         if (p) {
208:             switch (p->status) {
209:                 case 0x00: printf("XBee: txStatus: Success!\\n"); break;
210:                 case 0x01: printf("XBee: txStatus: No ACK\\n"); break;
211:                 case 0x02: printf("XBee: txStatus: CCA Failure\\n"); break;
212:                 case 0x03: printf("XBee: txStatus: Purged\\n"); break;
213:             }
214:             free(p);
215:         }
216:     }
217:     usleep(100000);
218: }
219:
220: return 0;
221: }

```