

```

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: #ifndef XBEE_H
22: #define XBEE_H
23:
24: #include <stdarg.h>
25:
26: enum xbee_types {
27:     xbee_unknown,
28:
29:     xbee_localAT, /* frame ID */
30:
31:     xbee_remoteAT,
32:     xbee_16bitRemoteAT, /* frame ID */
33:     xbee_64bitRemoteAT, /* frame ID */
34:
35:     xbee_16bitData, /* frame ID for ACKs */
36:     xbee_64bitData, /* frame ID for ACKs */
37:
38:     xbee_16bitIO,
39:     xbee_64bitIO,
40:
41:     xbee_txStatus,
42:     xbee_modemStatus
43: };
44: typedef enum xbee_types xbee_types;
45:
46: struct xbee_con {
47:     unsigned int tAddr64      : 1;
48:     unsigned int atQueue     : 1; /* queues AT commands until AC is sent */
49:     unsigned int txDisableACK : 1;
50:     unsigned int txBroadcast  : 1; /* broadcasts to PAN */
51:     unsigned int __spare__    : 4;
52:     xbee_types type;
53:     unsigned char frameID;
54:     unsigned char tAddr[8]; /* 64-bit 0-7 16-bit 0-1 */
55:     struct xbee_con *next;
56: };
57: typedef struct xbee_con xbee_con;
58:
59: struct xbee_sample {
60:     /* X A5 A4 A3 A2 A1 A0 D8 D7 D6 D5 D4 D3 D2 D1 D0 */
61:     unsigned short IOmask; /* IO */
62:     /* X X X X X X X D8 D7 D6 D5 D4 D3 D2 D1 D0 */
63:     unsigned short IOdigital; /* IO */
64:     /* X X X X X X D D D D D D D D D D */
65:     unsigned short IOanalog[6]; /* IO */
66: };
67: typedef struct xbee_sample xbee_sample;
68:
69: struct xbee_pkt {
70:     unsigned int sAddr64      : 1; /* yes / no */
71:     unsigned int dataPkt      : 1; /* if no - AT packet */
72:     unsigned int txStatusPkt  : 1;
73:     unsigned int modemStatusPkt : 1;
74:     unsigned int remoteATPkt   : 1;
75:     unsigned int IOPkt        : 1;
76:     unsigned int __spare__    : 2;
77:     xbee_types type;
78:     unsigned char frameID; /* AT Status */
79:     unsigned char atCmd[2]; /* AT */
80:     unsigned char status; /* AT Data Status */ /* status / options */
81:     unsigned char Addr64[8]; /* AT Data */
82:     unsigned char Addr16[2]; /* AT Data */
83:     unsigned char data[128]; /* AT Data */
84:     unsigned char RSSI; /* Data */
85:     unsigned int datalen;

```

```

86:
87:     struct xbee_pkt *next;
88:
89:     int samples;
90:     xbee_sample IOdata[1]; /* this array can be extended by using a this trick:
91:                             p = calloc(sizeof(xbee_pkt) + (sizeof(xbee_sample) * (samples - 1))) */
92: };
93: typedef struct xbee_pkt xbee_pkt;
94:
95: int xbee_setup(char *path, int baudrate);
96: int xbee_setuplog(char *path, int baudrate, int logfd);
97:
98: xbee_con *xbee_newcon(unsigned char frameID, xbee_types type, ...);
99:
100: void xbee_flushcon(xbee_con *con);
101:
102: #define xbee_endcon(x) xbee_endcon2((void **)&x)
103: void xbee_endcon2(xbee_con **con);
104:
105: int xbee_senddata(xbee_con *con, char *format, ...) __attribute__((format (printf,2,3)));
106: int xbee_vsnddata(xbee_con *con, char *format, va_list ap) __attribute__((format (printf,2,0)));
107: int xbee_nsenddata(xbee_con *con, char *data, int length);
108:
109: xbee_pkt *xbee_getpacketwait(xbee_con *con);
110: xbee_pkt *xbee_getpacket(xbee_con *con);
111:
112: int xbee_hasdigital(xbee_pkt *pkt, int sample, int input);
113: int xbee_getdigital(xbee_pkt *pkt, int sample, int input);
114:
115: int xbee_hasanalog(xbee_pkt *pkt, int sample, int input);
116: double xbee_getanalog(xbee_pkt *pkt, int sample, int input, double Vref);
117:
118: const char *svn_version(void);
119:
120: #endif

```