```c
/*
    libxbee - a C library to aid the use of Digi's Series 1 XBee modules
              running in API mode (AP=2).

    Copyright (C) 2009  Attie Grande (attie@attie.co.uk)

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
*/

#include "globals.h"

int main(int argc, char *argv[]) {
  xbee_con *con, *con2;
  xbee_pkt *pkt, *p;

  if (xbee_setup("/dev/ttyUSB0",57600) == -1) {
    perror("xbee_setup()");
    exit(1);
  }

  /*if ((con = xbee_newcon(NULL,'X',xbee_localAT)) == (void *)-1) {
    printf("error creating connection...\n");
    exit(1);
  }

  while(1){sleep(10);}

  xbee_senddata(con,"CH%c",0x0C);
  sleep(1);
  xbee_senddata(con,"ID%c%c",0x33, 0x32);
  sleep(1);
  xbee_senddata(con,"DH%c%c%c%c",0x00,0x00,0x00,0x00);
  sleep(1);
  xbee_senddata(con,"DL%c%c%c%c",0x00,0x00,0x00,0x00);
  sleep(1);
  xbee_senddata(con,"MY%c%c",0x00,0x00);
  sleep(1);
  // SH - read only
  // SL - read only
  xbee_senddata(con,"RR%c",0x00);
  sleep(1);
  xbee_senddata(con,"RN%c",0x00);
  sleep(1);
  xbee_senddata(con,"MM%c",0x00);
  sleep(1);
  xbee_senddata(con,"NT%c",0x19);
  sleep(1);
  xbee_senddata(con,"NO%c",0x00);
  sleep(1);
  xbee_senddata(con,"CE%c",0x00);
  sleep(1);
  xbee_senddata(con,"SC%c%c",0x1F,0xFE);
  sleep(1);
  xbee_senddata(con,"SD%c",0x04);
  sleep(1);
  xbee_senddata(con,"A1%c",0x00);
  sleep(1);
  xbee_senddata(con,"A2%c",0x00);
  sleep(1);
  // AI - read only
  xbee_senddata(con,"EE%c",0x00);
  sleep(1);
  //xbee_senddata(con,"KY%c",0x00);
  //sleep(1);
  xbee_senddata(con,"NI%s","TIGGER");
  sleep(1);
  xbee_senddata(con,"PL%c",0x04);
  sleep(1);
  xbee_senddata(con,"CA%c",0x2C);
  sleep(1);
  xbee_senddata(con,"SM%c",0x00);
  sleep(1);
  xbee_senddata(con,"ST%c%c",0x13,0x88);
  sleep(1);
```

```c
	xbee_senddata(con,"SP%c%c",0x00,0x00);
	sleep(1);
	xbee_senddata(con,"DP%c%c",0x03,0xE8);
	sleep(1);
	xbee_senddata(con,"SO%c",0x00);
	sleep(1);
	xbee_senddata(con,"BD%c",0x06);
	sleep(1);
	xbee_senddata(con,"RO%c",0x03);
	sleep(1);
	xbee_senddata(con,"AP%c",0x02);
	sleep(1);
	xbee_senddata(con,"PR%c",0xFF);
	sleep(1);
	xbee_senddata(con,"D8%c",0x00);
	sleep(1);
	xbee_senddata(con,"D7%c",0x01);
	sleep(1);
	xbee_senddata(con,"D6%c",0x00);
	sleep(1);
	xbee_senddata(con,"D5%c",0x01);
	sleep(1);
	xbee_senddata(con,"D4%c",0x00);
	sleep(1);
	xbee_senddata(con,"D3%c",0x00);
	sleep(1);
	xbee_senddata(con,"D2%c",0x00);
	sleep(1);
	xbee_senddata(con,"D1%c",0x00);
	sleep(1);
	xbee_senddata(con,"D0%c",0x00);
	sleep(1);
	xbee_senddata(con,"IU%c",0x00);
	sleep(1);
	xbee_senddata(con,"IT%c",0x01);
	sleep(1);
	xbee_senddata(con,"IC%c",0x00);
	sleep(1);
	xbee_senddata(con,"IR%c%c",0x00,0x00);
	sleep(1);
	xbee_senddata(con,"IA%c%c%c%c%c%c%c%c",0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF);
	sleep(1);
	xbee_senddata(con,"T0%c",0xFF);
	sleep(1);
	xbee_senddata(con,"T1%c",0xFF);
	sleep(1);
	xbee_senddata(con,"T2%c",0xFF);
	sleep(1);
	xbee_senddata(con,"T3%c",0xFF);
	sleep(1);
	xbee_senddata(con,"T4%c",0xFF);
	sleep(1);
	xbee_senddata(con,"T5%c",0xFF);
	sleep(1);
	xbee_senddata(con,"T6%c",0xFF);
	sleep(1);
	xbee_senddata(con,"T7%c",0xFF);
	sleep(1);
	xbee_senddata(con,"P0%c",0x01);
	sleep(1);
	xbee_senddata(con,"P1%c",0x00);
	sleep(1);
	xbee_senddata(con,"PT%c",0xFF);
	sleep(1);
	xbee_senddata(con,"RP%c",0x28);
	sleep(1);
	// VR - read only
	// HV - read only
	// DB - read only
	// EC - read only
	// EA - read only
	// DD - read only
	xbee_senddata(con,"CT%c",0x64);
	sleep(1);
	xbee_senddata(con,"GT%c%c",0x03,0xE8);
	sleep(1);
	xbee_senddata(con,"CC%c",0x2B);
	sleep(1);

	sleep(10);
	*/

	/* test 64bit IO and Data */
	con =  xbee_newcon('I',xbee_64bitIO,   0x0013A200, 0x403af247);
	con2 = xbee_newcon('I',xbee_64bitData, 0x0013A200, 0x403af247);
```

```c

  while (1) {
    while ((pkt = xbee_getpacket(con)) != NULL) {
      if (pkt->IOmask & 0x0001) printf("D0: %c  ",((pkt->IOdata & 0x0001)?'1':'0'));
      if (pkt->IOmask & 0x0002) printf("D1: %c  ",((pkt->IOdata & 0x0002)?'1':'0'));
      if (pkt->IOmask & 0x0004) printf("D2: %c  ",((pkt->IOdata & 0x0004)?'1':'0'));
      if (pkt->IOmask & 0x0008) printf("D3: %c  ",((pkt->IOdata & 0x0008)?'1':'0'));
      if (pkt->IOmask & 0x0010) printf("D4: %c  ",((pkt->IOdata & 0x0010)?'1':'0'));
      if (pkt->IOmask & 0x0020) printf("D5: %c  ",((pkt->IOdata & 0x0020)?'1':'0'));
      if (pkt->IOmask & 0x0040) printf("D6: %c  ",((pkt->IOdata & 0x0040)?'1':'0'));
      if (pkt->IOmask & 0x0080) printf("D7: %c  ",((pkt->IOdata & 0x0080)?'1':'0'));
      if (pkt->IOmask & 0x0100) printf("D8: %c  ",((pkt->IOdata & 0x0100)?'1':'0'));
#define Vref 3.23
      if (pkt->IOmask & 0x0200) printf("A0: %.2fv  ",(Vref/1024)*pkt->IOanalog[0]);
      if (pkt->IOmask & 0x0400) printf("A1: %.2fv  ",(Vref/1024)*pkt->IOanalog[1]);
      if (pkt->IOmask & 0x0800) printf("A2: %.2fv  ",(Vref/1024)*pkt->IOanalog[2]);
      if (pkt->IOmask & 0x1000) printf("A3: %.2fv  ",(Vref/1024)*pkt->IOanalog[3]);
      if (pkt->IOmask & 0x2000) printf("A4: %.2fv  ",(Vref/1024)*pkt->IOanalog[4]);
      if (pkt->IOmask & 0x4000) printf("A5: %.2fv  ",(Vref/1024)*pkt->IOanalog[5]);
      printf("\n");
      p = xbee_senddata(con2, "the time is %d\r", time(NULL));
      free(pkt);
      if (p) {
        switch (p->status) {
        case 0x01: printf("XBee: txStatus: No ACK\n");      break;
        case 0x02: printf("XBee: txStatus: CCA Failure\n"); break;
        case 0x03: printf("XBee: txStatus: Purged\n");      break;
        }
        free(p);
      }
    }
    while ((pkt = xbee_getpacket(con2)) != NULL) {
      printf("he said '%s'\n", pkt->data);
      p = xbee_senddata(con2, "you said '%s'\r", pkt->data);
      free(pkt);
      if (p) {
        switch (p->status) {
        case 0x01: printf("XBee: txStatus: No ACK\n");      break;
        case 0x02: printf("XBee: txStatus: CCA Failure\n"); break;
        case 0x03: printf("XBee: txStatus: Purged\n");      break;
        }
        free(p);
      }
    }
    usleep(100);
  }

  return 0;
}
```