```c
/*
 *  OMEGA CNi843 temperature monitor device support module
 *
 *  Why would it be a very bad idea to use this driver "as is"?
 */
#include    <vxWorks.h>
#include    <stdlib.h>
#include    <stdio.h>
#include    <string.h>
#include    <types.h>
#include    <socket.h>
#include    <in.h>
#include    <unistd.h>
#include    <float.h>
#include    <alarm.h>
#include    <cvtTable.h>
#include    <dbDefs.h>
#include    <dbAccess.h>
#include    <recSup.h>
#include    <devSup.h>
#include    <link.h>
#include    <aiRecord.h>

char monitor_IP[20]; /* Hint # 1 */

static long init_CNi843_record(struct aiRecord  *pai)
{
    struct instio *pinstio;

    /* For CNi843 module, ai.inp must be a INST_IO */
    switch (pai->inp.type)
    {
        case(INST_IO):
            pinstio = (struct instio *)&(pai->inp.value);
            strcpy(monitor_IP, pinstio->string);
            /* get ip address from db file */
            break;
        default:
            recGblRecordError(S_db_badField,(void *)pai,
                   "devAiCNi843 (init_record) Illegal INP field");
            return(S_db_badField);
    }

    return(0);
}

static long read_CNi843_ai(struct aiRecord  *pai)
{
    float value;

    Omega_temperature_read(monitor_IP, &value);
    pai->val = value;
    pai->udf = FALSE;
    return(2);  /* return raw data, don’t convert */
}

/* Hint # 2 */
void Omega_temperature_read(const char * ipaddress, float *temp)
{
    int fd;
    struct sockaddr_in address;
    int address_len;
    int rtval;
    short int port;
    char sendbuf[20];
    char recvbuf[20];
    char tmp[4];
    int len,i;

    port = 1000;

    /* Hint #3 */
    fd = socket(PF_INET, SOCK_STREAM, 0);
    address.sin_family = PF_INET;
    address.sin_addr.s_addr = inet_addr(ipaddress);
    address.sin_port = htons(port);
    address_len = sizeof(address);

    rtval = connect(fd, (struct sockaddr *)&address, address_len);
    if(rtval == -1)
        exit(1); /* Hint #4 */

    /*Send command first. */
    strcpy(sendbuf, "*01X01\r");
    send(fd, sendbuf, strlen(sendbuf), 0);

    /* Mega Hint #5 */

    /*Receive data from server. */
    len = recv(fd, recvbuf, 100, 0);

    /*get real data from responese message*/
    for(i = 0; i < 4; i++)
        tmp[i] = recvbuf[6+i];
    *temp = atof(tmp);

    /*Close and quit    */
    close(fd);
}

struct {
    long        number;
    DEVSUPFUN   report;
    DEVSUPFUN   init;
    DEVSUPFUN   init_record;
    DEVSUPFUN   get_ioint_info;
    DEVSUPFUN   read_ai;
    DEVSUPFUN   special_linconv;
}devAiCNi843={
    6,
    NULL,
    NULL,
    init_CNi843_record,
    NULL,
    read_CNi843_ai,
    NULL
};
```