

```

May 03, 04 9:07                serialw32.cpp                Page 1/3
/* -----
Application example for using Tserial
-----
Copyright @ 2001                Thierry Schneider
                                thierry@tetraedre.com
-----
Filename : sertest2.cpp
Author   : Thierry Schneider
Created  : April 8th 2001
Modified : April 24th 2001
Platform: Windows 95, 98, NT, 2000 (Win32)
-----
This software is given without any warranty. It can be distributed
free of charge as long as this header remains, unchanged.
-----
01.04.24  # ifdef __BORLANDC__ added in the header
----- */

/* ----- */
#ifdef __BORLANDC__
#pragma hdrstop                // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif

/* ----- */
#include "conio.h"
#include "tserial.hpp"

#include "serial.h"
FILE *fCode; FILE *fDATA;      // file pointer, only for testing on PC
char codeArray[1000][255];
int lineNum[1000];
int max;
int rowC=0, colC=0, rowD=0, colD=0;
Tserial *com;

#define DEBUG 0

// given a line number, find index less than or equal
int getIndex(int num) {
    int i;
    if(DEBUG) printf("\nSearch for %d: ", num);
    for(i=0;i<max;i++) {
        if(DEBUG) printf("%d.", lineNum[i]);
        if(lineNum[i]>=num) return i;
    }
    return -1;
}

int NewLine() {
    int num=0;

```

```

May 03, 04 9:07                serialw32.cpp                Page 2/3
    unsigned char byte;
    byte = ((com->getChar()&0xff));
    num|=(byte<<8);
    if(DEBUG) printf("Received byte: 0x%x, for %d\n", byte, num);
    byte = ((com->getChar()&0xff));
    num|=(byte);
    if(DEBUG) printf("Received byte: 0x%x, for %d\n", byte, num);
    return getIndex(num);
}

int main(int argc, char* argv[])
{
    int c;
    int len;
    int i=0;
//    int i;
//    int n, old_n;
    char fname[256];

    com = new Tserial();
    if (com!=0)
    {
        printf("Connecting on COM1 at 9600baud with no parity\n");
        com->connect("COM1", 9600, spNONE);

        while(1) {
            c = com->getChar();
            if(DEBUG) printf("received character (%x)\n", c);
            switch(c) {
                case SerialOpenFile:
                    if(DEBUG) printf("SerialOpenFile\n");
                    len=com->getChar();
                    if(DEBUG) printf("Size of filename is %d\n", len);
                    for(i=0;i<len;i++) {
                        fname[i]=com->getChar();
                        fname[i+1]='\0';
                        if(DEBUG) printf("new char = %c\n",
                            if(DEBUG) printf("fname= %s\n", fname);
                    }
                    if(DEBUG) printf("Done reading filename\n");
                    //com->getString(fname, c);
                    fCode = fopen(fname, "r");
                    fDATA = fopen(fname, "r");
                    if(fCode==NULL || fDATA==NULL) {
                        if(DEBUG) printf("Could not open file:
                            com->sendChar(SerialFailureOpenFile);
                    }else {
                        if(DEBUG) printf("Opened file: %s\n",
                            com->sendChar(SerialSuccessOpenFile);
                    }
                    for(rowC=0;fCode!=NULL;rowC++) {
                        fgets(codeArray[rowC], 255, fCode);
                        sscanf(codeArray[rowC], "%d", &lineNum[rowC]);
                        if(lineNum[rowC]==0) {max=rowC;rowC++;}
                    }
                    for(rowC=0;rowC<max;rowC++) {
                        if(DEBUG) printf(codeArray[rowC]

```

May 03, 04 9:07

serialw32.cpp

Page 3/3

```

        }
        rowC=0;rowD=0;colC=0;colD=0;
        break;
    case SerialNextChar:
        if(DEBUG) printf("SerialNextChar");
        //
        //
        c=getc(fCode);
        c=codeArray[rowC][colC++];
        if(DEBUG) printf("%c", c);
        com->sendChar(c);
        break;
    case SerialResetPointer:
        if(DEBUG) printf("SerialResetPointer");
        //
        //
        //
        fclose(fCode);
        fCode = fopen(fname, "r");
        if(DEBUG) printf("File %s reopened\n", f
name);

        rowC=NewLine();
        colC=0;
        if(DEBUG) printf("Reset (code) to line %d\n", ro
wC);

        break;
    case SerialNextCharDATA:
        if(DEBUG) printf("SerialNextCharDATA");
        //c=getc(fDATA);
        if(codeArray[rowD][colD]==0) {
            rowD++;
            colD=0;
            // after reset to new line, ther
e is no line number, then reset to begin
            if(codeArray[rowD][colD]<'0' ||
codeArray[rowD][colD]>'9') {rowD=0;colD=0;}
        }
        c=codeArray[rowD][colD++];
        if(DEBUG) printf("(%c)", c);
        com->sendChar(c);
        break;
    case SerialResetPointerDATA:
        if(DEBUG) printf("SerialResetPointerDATA");
        //
        //
        //
        fclose(fDATA);
        fDATA = fopen(fname, "r");
        if(DEBUG) printf("File %s reopened\n", f
name);
        //
        //
        rowD=NewLine();
        rowD=0;
        colD=0;
        if(DEBUG) printf("Reset (data) to line %d\n", ro
wD);

        break;
    case SerialRequestInput:
        if(DEBUG) printf("SerialRequestInput (from keybo
ard)");

        //c=fgetc(stdin);
        c=_getch();
        if(DEBUG) printf("Input Character %c\n", c);
        com->sendChar(c);
        break;
    default:
        printf("Command Not implemented\n");
        break;
    }
}
com->disconnect();
delete com;
com = 0;
}
return 0;
}

```