

Apr 28, 04 12:09

tserial.cpp

Page 1/4

```

/* -----
--
--          PC serial port connection object
--          for non-event-driven programs
--
--
-- Copyright @ 2001      Thierry Schneider
--                      thierry@tetraedre.com
--
-- -----
--
-- Filename : tserial.cpp
-- Author   : Thierry Schneider
-- Created  : April 4th 2000
-- Modified : April 8th 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.
-- ----- */

/* ----- */
#define STRICT
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <process.h>
#include <conio.h>
#include <windows.h>

#include "tserial.hpp"
#include "serial.h"

/* ----- */
/* ----- Tserial ----- */
/* ----- */
Tserial::Tserial()
{
    parityMode      = spNONE;
    port[0]         = 0;
    rate            = 0;
    serial_handle   = INVALID_HANDLE_VALUE;
}

/* ----- */
/* ----- ~Tserial ----- */
/* ----- */
Tserial::~Tserial()
{
    if (serial_handle!=INVALID_HANDLE_VALUE)
        CloseHandle(serial_handle);
    serial_handle = INVALID_HANDLE_VALUE;
}

/* ----- */
/* ----- disconnect ----- */
/* ----- */
void Tserial::disconnect(void)
{

```

Monday May 03, 2004

Apr 28, 04 12:09

tserial.cpp

Page 2/4

```

    if (serial_handle!=INVALID_HANDLE_VALUE)
        CloseHandle(serial_handle);
    serial_handle = INVALID_HANDLE_VALUE;
}

/* ----- */
/* ----- connect ----- */
/* ----- */
int Tserial::connect      (char *port_arg, int rate_arg, serial_parity pari
ty_arg)
{
    int erreur;
    DCB dcb;
    COMMTIMEOUTS cto = { 0, 0, 0, 0, 0 };

    /* ----- */
    if (serial_handle!=INVALID_HANDLE_VALUE)
        CloseHandle(serial_handle);
    serial_handle = INVALID_HANDLE_VALUE;

    erreur = 0;

    if (port_arg!=0)
    {
        strncpy(port, port_arg, 10);
        rate      = rate_arg;
        parityMode= parity_arg;
        memset(&dcb,0,sizeof(dcb));

        /* ----- */

        /* set DCB to configure the serial port
        dcb.DCBlength      = sizeof(dcb);

        /* ----- Serial Port Config ----- */
        dcb.BaudRate      = rate;

        switch(parityMode)
        {
            case spNONE:
                dcb.Parity      = NOPARITY;
                dcb.fParity     = 0;
                break;

            case spEVEN:
                dcb.Parity      = EVENPARITY;
                dcb.fParity     = 1;
                break;

            case spODD:
                dcb.Parity      = ODDPARITY;
                dcb.fParity     = 1;
                break;

        }

        dcb.StopBits      = ONESTOPBIT;
        dcb.ByteSize      = 8;

        dcb.fOutxCtsFlow   = 0;
        dcb.fOutxDsrFlow  = 0;
        dcb.fDtrControl    = DTR_CONTROL_DISABLE;
        dcb.fDsrSensitivity = 0;
        dcb.fRtsControl    = RTS_CONTROL_DISABLE;
        dcb.fOutX          = 0;
        dcb.fInX           = 0;

        /* ----- misc parameters ----- */
        dcb.fErrorChar     = 0;
        dcb.fBinary        = 1;
        dcb.fNull          = 0;
        dcb.fAbortOnError  = 0;

```

tserial.cpp

1/2

Apr 28, 04 12:09 **tserial.cpp** Page 3/4

```

dcb.wReserved      = 0;
dcb.XonLim         = 2;
dcb.XoffLim        = 4;
dcb.XonChar        = 0x13;
dcb.XoffChar       = 0x19;
dcb.EvtChar        = 0;

/* -----
*/
serial_handle = CreateFile(port, GENERIC_READ | GENERIC_WRITE,
                          0, NULL, OPEN_EXISTING, NULL, NULL);
// opening serial port

if (serial_handle != INVALID_HANDLE_VALUE)
{
    if (!SetCommMask(serial_handle, 0))
        erreur = 1;

    // set timeouts
    if (!SetCommTimeouts(serial_handle, &cto))
        erreur = 2;

    // set DCB
    if (!SetCommState(serial_handle, &dcb))
        erreur = 4;
}
else
    erreur = 8;
}
else
    erreur = 16;

/* ----- */
if (erreur != 0)
{
    CloseHandle(serial_handle);
    serial_handle = INVALID_HANDLE_VALUE;
}
return(erreur);
}

/* ----- */
/* ----- sendChar ----- */
/* ----- */
void Tserial::sendChar(char data)
{
    sendArray(&data, 1);
}

/* ----- */
/* ----- sendArray ----- */
/* ----- */
void Tserial::sendArray(char *buffer, int len)
{
    unsigned long result;

    if (serial_handle != INVALID_HANDLE_VALUE)
        WriteFile(serial_handle, buffer, len, &result, NULL);
}

#ifdef WINDEBUG 1
const int MAXCMDS=30;
int cmds[MAXCMDS]={SerialOpenFile, 14, 't', 'e', 's', 't', '/', 'g', 'u', 'e', 's',
's', '.', 'b', 'a', 's', SerialResetPointer, 0x0, 0x0, 0x4, 0x10, SerialNextChar,
SerialNextChar};
int cmd=0;

```

Apr 28, 04 12:09 **tserial.cpp** Page 4/4

```

/* ----- */
/* ----- getChar ----- */
/* ----- */
char Tserial::getChar(void)
{
#ifdef WINDEBUG
    if (cmd < MAXCMDS) return cmds[cmd++];
    else while(1);
#else
    char c;
    getArray(&c, 1);
    return(c);
#endif
}

/* ----- */
/* ----- getArray ----- */
/* ----- */
int Tserial::getArray(char *buffer, int len)
{
    unsigned long read_nbr;

    read_nbr = 0;
    if (serial_handle != INVALID_HANDLE_VALUE)
    {
        ReadFile(serial_handle, buffer, len, &read_nbr, NULL);
    }
    return((int) read_nbr);
}

//added by David Hodgdon
/* ----- */
/* ----- getString ----- */
/* ----- */
int Tserial::getString(char *buffer, int len)
{
    int returnVal;
    returnVal = getArray(buffer, len);
    buffer[len]='\0';
    return returnVal;
}

//end added by David Hodgdon

/* ----- */
/* ----- getNbrOfBytes ----- */
/* ----- */
int Tserial::getNbrOfBytes(void)
{
    struct _COMSTAT status;
    int n;
    unsigned long etat;

    n = 0;

    if (serial_handle != INVALID_HANDLE_VALUE)
    {
        ClearCommError(serial_handle, &etat, &status);
        n = status.cbInQue;
    }

    return(n);
}

```