```
// Projekt        Patient overvågning
//
// Fil            SerialCom.cpp
//
// Beskrivelse    Implementering af klassen SerialCom.
//                En klasse til serial kommunikation.
//
// Forfatter      Erik Gross Jensen
//
// Version        1.0 041200 EGJ - oprindelig version
//                1.1 040202 EGJ - flere kommentarer tilføjet
//                1.2 181108 NVJ - tilrettet 1. sem. projekt

#include "SerialCom.h"


//**********************************************************************
bool SerialCom::open( int port, int baud )
// Åbner en seriel port for kommunikation.
// Input:  port : Comport nummer 1 eller 2
//         baud : Hastigheden for kommunikationen: 9600, 19200 eller 38400
// Output: true hvis porten kunne åbnes
//         false hvis porten ikke kunne åbnes. Når der returneres false skyldes
//         det ofte at et andet program bruger den serielle port
//**********************************************************************
{
    char portstr[5] = "COM ";

    portstr[3] = 48 + port ;


    HComdev = CreateFile( portstr,  // Navnet på den port der skal åbnes(COMX)
                          GENERIC_READ | GENERIC_WRITE,  // read/write types
                          NULL,
                          0,
                          OPEN_EXISTING,
                          0,
                          0);

    if( HComdev == INVALID_HANDLE_VALUE )
        return false;
    else
    {
        //sæt hastighed m.m.
        GetCommState( HComdev, &dcb );
        dcb.DCBlength = sizeof(DCB);

        //set baud rate
        if( baud == 9600 )
            dcb.BaudRate = CBR_9600;
        else if( baud == 19200 )
            dcb.BaudRate = CBR_19200;
        else if( baud == 38400 )
            dcb.BaudRate = CBR_38400;
        else
            return false;

        // set databit
        dcb.ByteSize = 8;

        //set paritet
        dcb.Parity = 0;

        //set stopbit
        dcb.StopBits = ONESTOPBIT;

        SetCommState( HComdev, &dcb );

        return true;
    }
}
```

```
//*****************************************
bool SerialCom::close()
// Lukker forbindelsen
// Input:
// Output: true hvis forbindelsen blev lukket ellers false
//*****************************************
{
    if( CloseHandle(HComdev) )
        return true;
    else
        return false;
}


//************************************************
bool SerialCom::send( char *sendPtr, int antal )
// Sender et antal karakterer på en seriel port.
// Husk porten skal være åben inden send kan bruges
// Input: *sendPtr : en pointer til den char streng der skal sendes
//        antal    : Er det antal char der skal sendes
// Output: true hvis afsendelse gik godt
//         false hvis der ikke blev sendt noget. Når der returneres
//         false skyldes det ofte at porten ikke er åbnet
//************************************************
{
    DWORD dwBytesWritten;

    if( WriteFile(HComdev, sendPtr, antal, &dwBytesWritten, 0) )
        return true ;
    else
        return false;
}


//************************************************
int SerialCom::inWatingBuffer()
// Tæller antallet af karakterer der findes i receive buffer
// Husk porten skal være åben inden send kan bruges
// Input:
// Output: returnerer antallet af karakterer der er findes i receive buffer
//************************************************
{
    ClearCommError( HComdev, &dwErrorFlags, &ComStat );
    return ComStat.cbInQue ;
}


//************************************************
int SerialCom::receive( char *rxPtr )
// Henter de karakterer der findes i receive buffer. Bemærk hvis der
// ingen karakterer er i receive buffer vil funktionen først returnere
// når der kommer en karakter i  receive bufferen.
// Input: en pointer til et char array hvor data skal overføres til
// Output: returnerer det antal karakterer der er overført til rxPtr
//************************************************
{
    DWORD byteRead,dwBytesRead;

    dwBytesRead = inWatingBuffer();
    ReadFile( HComdev, rxPtr, dwBytesRead, &byteRead, 0 );
    return byteRead;
}


//************************************************
char SerialCom::receiveOneChar()
// Læser 1 karakter fra receive buffer. Bemærk hvis der ingen karakterer
// er i receive buffer vil funktionen først returnere, når der kommer en
// karakter i receive bufferen.
// Input:
// Output: Returnerer den læste karakter.
//************************************************
{
```

```
    char rxBuf;
    DWORD byteRead;

    ReadFile( HComdev, &rxBuf, 1, &byteRead, 0 );
    return rxBuf;
}
```