Search the Arduino Playground

Manuals and Curriculum
(http://playground.arduino.cc/Main/ManualsAndCurriculum)
Arduino StackExchange
(http://arduino.stackexchange.com)
Board Setup and Configuration
(http://playground.arduino.cc/Main/ArduinoCoreHardware)
Development Tools
(http://playground.arduino.cc/Main/DevelopmentTools)
Arduino on other Atmel Chips
(http://playground.arduino.cc/Main/ArduinoOnOtherAtmelChips)
Interfacing With Hardware
(http://playground.arduino.cc/Main/InterfacingWithHardware)

- Output
  (http://playground.arduino.cc/Main/InterfacingWithHardware#Output)
- Input
  (http://playground.arduino.cc/Main/InterfacingWithHardware#InputTOC)
- User Interface
  (http://playground.arduino.cc/Main/InterfacingWithHardware#ui)
- Storage
  (http://playground.arduino.cc/Main/InterfacingWithHardware#Storage)
- Communication
  (http://playground.arduino.cc/Main/InterfacingWithHardware#Communication)
- Power supplies
  (http://playground.arduino.cc/Main/IntWithHW-PwrSup)
- General
  (http://playground.arduino.cc/Main/InterfacingWithHardware#General)

Interfacing with Software
(http://playground.arduino.cc/Main/InterfacingWithSoftware)
User Code Library
(http://playground.arduino.cc/Main/GeneralCodeLibrary)

- Snippets and Sketches
  (http://playground.arduino.cc/Main/SketchList)
- Libraries
  (http://playground.arduino.cc/Main/LibraryList)
- Tutorials
  (http://playground.arduino.cc/Main/TutorialList)

Suggestions & Bugs
(http://code.google.com/p/arduino/issues/list)

Electronics Technique (http://playground.arduino.cc/Main/ElectroInfoResources)

Sources for Electronic Parts (http://playground.arduino.cc/Main/Resources)

Related Hardware and Initiatives (http://playground.arduino.cc/Main/SimilarBoards)

Arduino People/Groups & Sites (http://playground.arduino.cc/Main/People)

Exhibition (http://playground.arduino.cc/Projects/ArduinoUsers)

Project Ideas (http://playground.arduino.cc/Projects/Ideas)

Languages (http://playground.arduino.cc/Main/Languages)

PARTICIPATE (http://playground.arduino.cc/Main/Participate)

- Suggestions (http://code.google.com/p/arduino/issues/list)
- Formatting guidelines (http://playground.arduino.cc/Main/Participate#contribrules)
- All recent changes (http://playground.arduino.cc/Site/AllRecentChanges)
- PmWiki (http://playground.arduino.cc/PmWiki/PmWiki)
- WikiSandBox training (http://playground.arduino.cc/Main/WikiSandbox)
- Basic Editing (http://playground.arduino.cc/PmWiki/BasicEditing)
- Cookbook (addons) (http://www.pmwiki.org/wiki/Cookbook/CookbookBasics)
- Documentation index (http://www.pmwiki.org/wiki/PmWiki/DocumentationIndex)
- Drone with Arduino (http://www.bartoloilliano.com/arduino-tutorial-costruire-un-drone-con-webcam-telecomandato-da-pc-tramite-csharp.html)
- Thermostat with Arduino (http://arduinothermostat.blogspot.co.uk)

Interfacing... (http://playground.arduino.cc/Main/Interfacing)

Arduino and C++ (for Windows)

As I found it pretty hard finding the good information, or an already working code to handle Serial communication on windows based system, I finally made a class that do what is needed for basic Serial Communication, thanks to help of several forumers. Please note that this code might not be completely perfect so I encourage you to make any update needed so that it might become even better.

So now for the code which is consisting of two files, a header and a source code file.

Click here (http://playground.arduino.cc/Interfacing/CPPWindows#VisualStudio2008) for an easy example using Visual Studio 2008 Express.

Important note: when connecting to some Arduinos, it is necessary to use COM port "device names" as outlined at http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1286253977/18#18 (http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1286253977/18#18).

Furthermore: When addressing ports larger than COM9 in Windows you will have to specify the port thusly: "COM10" becomes "\\\\.\\COM10" See: http://support.microsoft.com/default.aspx?scid=kb;EN-US;q115831 (http://support.microsoft.com/default.aspx?scid=kb;EN-US;q115831)

And: CreateFile may need to be replaced with CreateFileA in Serial.cpp

SerialClass.h (header)

```
#ifndef SERIALCLASS_H_INCLUDED
#define SERIALCLASS_H_INCLUDED

#define ARDUINO_WAIT_TIME 2000

#include <windows.h>
#include <stdio.h>
#include <stdlib.h>

class Serial
{
    private:
        //Serial comm handler
        HANDLE hSerial;
        //Connection status
        bool connected;
        //Get various information about the connection
        COMSTAT status;
        //Keep track of last error
        DWORD errors;

    public:
        //Initialize Serial communication with the given COM port
        Serial(char *portName);
        //Close the connection
        //NOTA: for some reason you can't connect again before exiting
        //the program and running it again
        ~Serial();
        //Read data in a buffer, if nbChar is greater than the
        //maximum number of bytes available, it will return only the
        //bytes available. The function return -1 when nothing could
        //be read, the number of bytes actually read.
        int ReadData(char *buffer, unsigned int nbChar);
        //Writes data from a buffer through the Serial connection
        //return true on success.
        bool WriteData(char *buffer, unsigned int nbChar);
        //Check if we are actually connected
        bool IsConnected();


};

#endif // SERIALCLASS_H_INCLUDED
```

Serial.cpp (source code file)

```
#include "SerialClass.h"

Serial::Serial(char *portName)
{
    //We're not yet connected
    this->connected = false;

    //Try to connect to the given port throuh CreateFile
    this->hSerial = CreateFile(portName,
            GENERIC_READ | GENERIC_WRITE,
            0,
            NULL,
            OPEN_EXISTING,
            FILE_ATTRIBUTE_NORMAL,
            NULL);

    //Check if the connection was successfull
    if(this->hSerial==INVALID_HANDLE_VALUE)
    {
        //If not success full display an Error
        if(GetLastError()==ERROR_FILE_NOT_FOUND){

            //Print Error if neccessary
            printf("ERROR: Handle was not attached. Reason: %s not available.\n", portName);

        }
        else
        {
            printf("ERROR!!!");
        }
    }
    else
    {
        //If connected we try to set the comm parameters
        DCB dcbSerialParams = {0};

        //Try to get the current
        if (!GetCommState(this->hSerial, &dcbSerialParams))
        {
            //If impossible, show an error
            printf("failed to get current serial parameters!");
        }
        else
        {
            //Define serial connection parameters for the arduino board
            dcbSerialParams.BaudRate=CBR_9600;
            dcbSerialParams.ByteSize=8;
            dcbSerialParams.StopBits=ONESTOPBIT;
            dcbSerialParams.Parity=NOPARITY;

             //Set the parameters and check for their proper application
             if(!SetCommState(hSerial, &dcbSerialParams))
             {
                printf("ALERT: Could not set Serial Port parameters");
             }
             else
             {
                 //If everything went fine we're connected
                 this->connected = true;
                 //We wait 2s as the arduino board will be reseting
                 Sleep(ARDUINO_WAIT_TIME);
             }
        }
    }

}

Serial::~Serial()
{
    //Check if we are connected before trying to disconnect
    if(this->connected)
    {
        //We're no longer connected
        this->connected = false;
        //Close the serial handler
        CloseHandle(this->hSerial);
    }
}
```

```
int Serial::ReadData(char *buffer, unsigned int nbChar)
{
    //Number of bytes we'll have read
    DWORD bytesRead;
    //Number of bytes we'll really ask to read
    unsigned int toRead;

    //Use the ClearCommError function to get status info on the Serial port
    ClearCommError(this->hSerial, &this->errors, &this->status);

    //Check if there is something to read
    if(this->status.cbInQue>0)
    {
        //If there is we check if there is enough data to read the required number
        //of characters, if not we'll read only the available characters to prevent
        //locking of the application.
        if(this->status.cbInQue>nbChar)
        {
            toRead = nbChar;
        }
        else
        {
            toRead = this->status.cbInQue;
        }

        //Try to read the require number of chars, and return the number of read bytes on success
        if(ReadFile(this->hSerial, buffer, toRead, &bytesRead, NULL) && bytesRead != 0)
        {
            return bytesRead;
        }

    }

    //If nothing has been read, or that an error was detected return -1
    return -1;

}


bool Serial::WriteData(char *buffer, unsigned int nbChar)
{
    DWORD bytesSend;

    //Try to write the buffer on the Serial port
    if(!WriteFile(this->hSerial, (void *)buffer, nbChar, &bytesSend, 0))
    {
        //In case it don't work get comm error and return false
        ClearCommError(this->hSerial, &this->errors, &this->status);

        return false;
    }
    else
        return true;
}

bool Serial::IsConnected()
{
    //Simply return the connection status
    return this->connected;
}
```

That's all you need, it has been tested with CodeBlocks and the MinGW compiler but should work fine with Visual Studio too. As for the way to use it I think the comments of the header makes it quite simple to understand though if needed you can contact me on the forum.

Visual Studio 2008 Express Edition

This example builds on the serial_read_blink.pde sketch that is included in the Arduino + C example on the Interfacing With Software (http://playground.arduino.cc/Main/InterfacingWithSoftware) page. You need to have Visual Studio 2008 C++ Express Edition or the full version of Visual Studio 2008 C++ installed on your computer.

1. Download and Install Visual Studio 2008 Express Edition
2. File >> New >> Project...
3. "Visual C++" >> "CLR"
4. Select "Windows Form Application". Enter a name for your project. Click "OK"
5. Toolbox (right side of the screen) >> Components >> SerialPort. Click, Drag and then Drop "SerialPort" onto your form. It will show up as a resource at the bottom of the form window. You may optionally rename "serialport1" to "arduino" and change the code snippets below.
6. Toolbox >> Common Controls >> Button. Click, Drag and then Drop "Button" onto the form.
7. Double-Click on your new button. Enter the following code:

```
     this->serialPort1->PortName = "COM5";  // Replace with your COM port!
     this->serialPort1->Open();
     this->serialPort1->Write( "7" );  // In the future,  you'll expand on this
                                       // to write your custom data to the board
     this->serialPort1->Close();
```

1. Compile and run the program. Click on your button. The LED on the Arduino board should blink the same number of times as the number you entered in the Write statement above, only if it's from 1-9.
   Note: serialPort1 is the default name of the object, and if you change it, you'll need to change this code.

Example main.cpp for Windows

```
#include <stdio.h>
#include <tchar.h>
#include "SerialClass.h"        // Library described above
#include <string>

// application reads from the specified serial port and reports the collected data
int _tmain(int argc, _TCHAR* argv[])
{
        printf("Welcome to the serial test app!\n\n");

        Serial* SP = new Serial("\\\\.\\COM10");    // adjust as needed

        if (SP->IsConnected())
                printf("We're connected");

        char incomingData[256] = "";                    // don't forget to pre-allocate memory
        //printf("%s\n",incomingData);
        int dataLength = 256;
        int readResult = 0;

        while(SP->IsConnected())
        {
                readResult = SP->ReadData(incomingData,dataLength);
                printf("Bytes read: (-1 means no data available) %i\n",readResult);

                std::string test(incomingData);

                printf("%s",incomingData);
                test = "";

                Sleep(500);
        }
        return 0;
}
```

Share

NEWSLETTER

Enter your email to sign up →

Contact us (http://arduino.cc/en/Main/ContactUs)

(https://twitter.com/arduino) (http://www.facebook.com/official.arduino) (https://plus.google.com/+Arduino) (http://www.flickr.com/photos/arduino_cc)

(http://youtube.com/arduinoteam)