

- [Home](#)
- [About me](#)
- [Downloads](#)

THIJS.ELENBAAS.NET



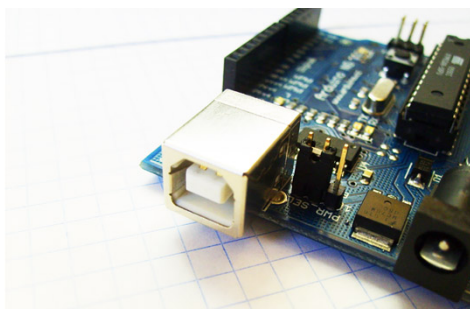
Arduino-to-PC messaging

September 7, 2013 by Thijs Elenbaas | [63 Comments](#)

Introduction

In our office, drinking espresso has become something of a science. We have tried many kinds of beans, mills, and even de-mineralized and custom re-mineralized water. The obvious next step is to modify our Gaggia Classic espresso machine for better temperature control. As a start I first want to measure and control the temperature of a water cooker using a thermocouple, and log the temperature to my PC. The next step would be to use this nice [PID](#) library (the [articles](#) describing it are extremely good, and even interesting if you just want to learn about PID control in general).

Since the serial-over-USB connection is the de-facto connection to the Arduino, I had expected to find a messaging protocol over the serial port without any issue. This turned out to be more difficult than I had expected. On the [Interfacing with software](#) page on the Arduino Playground, I found two mature, open and well maintained libraries, [Bitlash](#) and [Firmata](#), which seemed to be promising at first. However, they turned out to be not quite what I was looking for: both libraries are designed to place all Arduino control at the PC side and make the Arduino into a IO component. All processing would then happen on the PC side (for example in the [Processing](#) environment), instead of on the Arduino processor. In our case the end goal is that the Arduino works stand-alone to control the temperature, without needing an outside connection.



Arduino USB serial connection

When looking at library that does messaging only, I found the best option to be CmdMessenger 2. But even though it was already a pretty good piece of work, it was still lacking in some aspects:

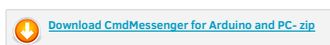
1. No CmdMessenger library for the PC side. For a messaging protocol to really work, both sides need to speak it.
2. A asymmetrical implementation. That is, while it is possible to receive all basic data-types (char arrays, floats, ints, bytes), it was limited to sending just strings.
3. No ability to send binary data. That is to say that some special characters, would be interpreted by the library to be field separators or command separators.
4. No ability to wait for a specific response from the other side

I added these features in this 3rd iteration of the CmdMessenger library. I believe that the only thing messaging protocols [should probably have](#) and that is missing is check-summing of commands. Then again, in most cases the serial protocol is over USB, which makes the chance of data corruption small. I seem to be the fourth author who worked on this library, which is unusual in Arduino land. Typically there is only one author to a library.

I think this has been the first time I implemented the same functionality in both C++ and C# (it works on .net and Mono), and it made me appreciate all the more how different these programming languages are, even though the syntax is quite similar. One of the reasons is that on the Arduino one typically programs ansi C, poured in classes. (there is no space for the STL libraries, dynamic arrays etc). The Arduino implementation of the library is quite tight, and uses only a small fraction of the resources that the C# version uses (clock cycles, program space, memory). On the other hand, the Arduino version does not support arbitrarily large buffers, multi-threading and all kinds of convenient tooling.

The library

The library can be downloaded here:



Or at the Github repository:

<https://github.com/thijse/Arduino-Libraries/tree/master/CmdMessenger>

An overview of all articles on CmdMessenger can be found here:

<http://thijs.elenbaas.net/tag/CmdMessenger/>

It comes with the Arduino and C# implementation, automatically generated documentation, and different samples of Arduino – PC communication, starting with basic command sending, and ending with advanced multi-parameter binary commands. Each sample consists of a Arduino sketch and a [Visual Studio/ MonoDevelop](#) project, designed to interact with each other.

Let's have a look at one example. In this example the PC will toggle the on-board led of the Arduino, by sending commands to set the state of the led. The Arduino receives, the command, reads the parameter with the requested led state and set the led accordingly. Next, the Arduino will send a command to the PC returning the set state.

First we need to define the commands. At Arduino side:

```
1 enum
2 {
3     kSetLed, // Command to request led to be set in specific state
4     kStatus, // Command to report status
5 };
```

And at the PC side:

```
1 enum Command
2 {
3     SetLed, // Command to request led to be set in specific state
4     Status, // Command to report status
5 };
```

Note that these should be the exact same commands, in the same order. Of course you can follow language dependent coding conventions, resulting in (slightly) different naming. On the Arduino side we need to listen to the SetLed command. We do this by attaching a callback function to this command ID:

```
1 void attachCommandCallbacks()
2 {
3     // Attach callback methods
4     cmdMessenger.attach(kSetLed, OnSetLed);
5 }
6 void OnSetLed()
7 {
8     // Read led state argument, interpret string as boolean
9     ledState = cmdMessenger.readBoolArg();
10    // Set led
11    digitalWrite(kBlinkLed, ledState?HIGH:LOW);
12    // Send back status that describes the led state
13    cmdMessenger.sendCmd(kStatus,(int)ledState);
14 }
```

On the PC side we will listen to the response with the Status command :

```
1 private void AttachCommandCallbacks()
2 {
3     _cmdMessenger.Attach((int)Command.Status, OnStatus);
4 }
5
6 // Callback function that prints the Arduino status to the console
7 void OnStatus(ReceivedCommand arguments)
8 {
9     Console.WriteLine("Arduino status: ");
10    Console.WriteLine(arguments.ReadStringArg());
11 }
```

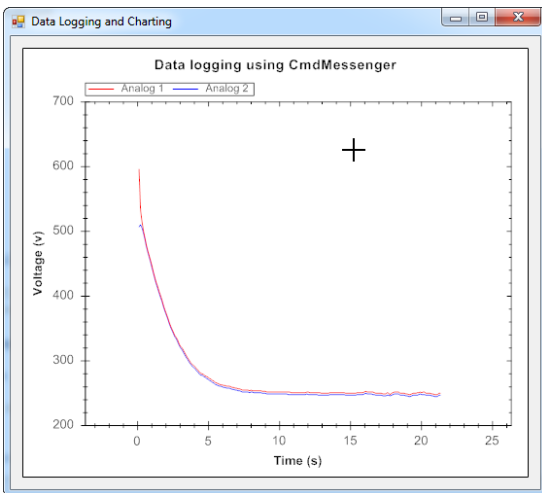
That's all there's to it for basic 2-way communication! As said before, the download package comes with more examples that show multiple arguments, binary arguments, waiting for acknowledgement, logging of sent and received data and more, as well as some [doxygen](#) generated documentation for both platforms.

So, please play around with it, and if you put it to good use, please leave a comment.

Update

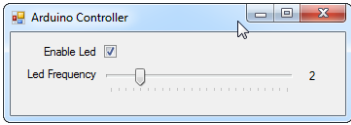
I had to make a few changes to the library to make it play nicely with applications that have a GUI (WinForms). I've also added two examples how to make a small GUI application.

The first example receives data from the Arduino AnalogPins and plots this data in real-time to a moving-window chart.



Log and Chart real time data from Arduino.

The second example allows you to change the blink frequency of the on-board led using a slider.



Control Arduino blink led using slider and check-box.

As the rest of the library, they work both for .NET and Mono.

Categories: [General](#) | Tags: [Arduino](#), [C++](#), [CmdMessenger](#), [code](#), [messaging](#), [Serial](#), [software communication](#) | [Permalink](#)

63 Comments

[Leave a reply](#) →

1.

Erik Jan Laret

[September 15, 2013 at 8:29 PM](#)

I had a lot of fun trying out the examples. As a newby in c# I failed however in incorporating this in an application with a form. Could you provide an example in c# using a form with one button switching on/off the LED.

That would show me where the usings, constructors etc go.

Thanks

Erik Jan Laret

[Reply](#)



o

[Thijs Elenbaas](#)

[September 25, 2013 at 1:36 PM](#)

Good to hear you had fun!

I'm in the process of writing and documenting another two examples, which will explain how to use the CmdMessenger library with WinForms applications. The first one receives data from the Arduino AnalogPins and plots that real-time in a moving-window chart. The second one allows you to change the blink frequency of the on-board led using a slider.

I'll keep you posted!

[Reply](#)

■

Erik Jan

[October 4, 2013 at 6:33 PM](#)

That would be great, it like you read my mind, a rolling graph is exactly what I want to do!. Look forward to your examples.

[Reply](#)



■

[Thijs Elenbaas](#)

[April 7, 2014 at 9:14 AM](#)

Hi Erik Jan,

And? Are they working for you?

[Reply](#)

2.

Gary

[September 17, 2013 at 9:39 PM](#)

Hey thanks for the article. I couldn't get the GITHUB link to work for some reason though?

[Reply](#)



o

[Thijs Elenbaas](#)

[September 25, 2013 at 2:02 PM](#)

Thanks, I'll check the link. For now, the Github repository can be found here: <https://github.com/thijse/Arduino-Libraries>

[Reply](#)

3.

LeRoyCorbid

[September 22, 2013 at 12:14 AM](#)

Thanks for the work,

Been Playing with it for a couple of days and I think I got it figured out.

If I can realize my vision my App will send a command char code to Duino, Duino will respond by either sending three float values representing Ping Sensor results, an int analog value from a pot representing a speed setting, or one of four movement actions. If movement arduino will send an int Ack value to confirm, else PC will ack the values sent to it..

I am a very new to coding and this is my first project using c# (love it!!!)

Wish me luck and Thanks again.

[Reply](#)



o [Thijs Elenbaas](#)
[September 25, 2013 at 1:33 PM](#)

First of all: Good luck!

I personally like C# better than all other languages I've used: the familiarity of C++, but without the hassle of header files and adding nice new functionality such as events and lambda functions. The garbage collector (automatic disposal of unused objects) is a bit of a mixed bag, though. It is very easy for smaller projects, but for larger, memory hungry projects, you might want to have full control over object disposal.

Anyway...

Your program sounds interesting, but I'm not sure what the overall goal of it is. If you have more information (screenshots), please post them here and/or on the Arduino forum!

Thanks!

ps. I'm writing a few WinForm samples (see other comment) that may help you along further.

[Reply](#)

4.

[Jasper](#)
[September 26, 2013 at 10:48 PM](#)

this looks like exactly what we need for a commercial project, we are going to try it out and if it is suitable, how much would it cost to purchase a "developer's licence" from you? or perhaps pay you royalties when we start selling our product?

[Reply](#)



o [Thijs Elenbaas](#)
[September 27, 2013 at 9:07 AM](#)

Hi Jasper,

Well, that's a nice compliment! The code has been released under the MIT license, which is a very liberal license (as well as very simple, this is the full license: <http://opensource.org/licenses/mit-license.html>). This means that you can use the code in your commercial product, as long as you keep the copyright notice and permission notice intact.

Nevertheless, I am enthusiastic about CmdMessenger, and I am contemplating some features such as:

- (Collapsible) command queues for both sending and receiving. This will solve possible congestion and deadlocks
- Auto connect, that is, scanning the parallel ports and possible bit rates until a responding CmdMessenger is found
- Unit testing

So, maybe we can find a way in which you can support the project and help each other. I'll send you an email

[Reply](#)

5.

[Joost](#)
[September 29, 2013 at 9:12 AM](#)

Hi Thijs,

Great library! I found it easy to set up in Windows using Visual Studio. However my end goal is for my Synology NAS, which runs Linux, to communicate with the Arduino. I deduce from your website you have one too. Have you ever tried this? Could you give me any tips? I'm familiar with programming in Windows but not so much in Linux. I think it can be done in c++, but a code snippet

[Reply](#)



o [Thijs Elenbaas](#)
[September 30, 2013 at 9:29 AM](#)

Hi, Joost. Luckily there is a mono package for the Synology, see http://www.synology.com/releaseNote_en/package_Mono.php?lang=enu. The other thing you need is a serial port over USB to connect to the Arduino. You could try this: <http://forum.synology.com/enu/viewtopic.php?f=27&t=14813>

Regarding working under Linux, I would advise you to write your Synology-side program in a Linux environment (like [Ubuntu](#)), using [MonoDevelop for Linux](#). This gives you a much more development friendly environment than the Synology itself. Only when your program works, take it to your NAS. If you don't have a PC on which you want to install Linux, try [running it in a Virtual Machine](#)

Out of curiosity, what are you working on?

[Reply](#)

■ [Joost](#)
[September 30, 2013 at 1:04 PM](#)

Thx for the very elaborate reply! I'm working on making a smart thermostat, which I can turn on/off remotely using my iPhone. For this to happen a lot of steps need to be automated and communication between a webserver (with webservice) and the arduino is one. This is the one I'm currently working on.

[Reply](#)



■
[Thijs Elenbaas](#)
[September 30, 2013 at 1:54 PM](#)

Sounds like an interesting project! Mono supports ASP.NET so it is pretty easy to [make it into a web service](#). You could use the webserver that runs on your Synology, but there is a risk that you screw up your NAS webconsole. You can also host your website using [XSP](#).

Have you considered buying a [raspberry Pi](#) and install [Raspbian](#) and MonoDevelop on it? Cheap, much better development environment, no chance of breaking your NAS.

[Reply](#)

■
[Joost](#)
[September 30, 2013 at 2:16 PM](#)

Hi,

I actually own a raspberry Pi, which I use as a mediacenter using OpenElec XBMC. It might be worth a shot buying a second SD card and installing Raspbian with Monodevelop on it. Thanks for the tip! I'll keep you posted on my progress...

6.
[Joost](#)
[September 29, 2013 at 9:14 AM](#)

Mmm... Stupid iPhone. In addition to previous comment: would be great!

[Reply](#)

7.
[xchg.ca](#)
[September 30, 2013 at 8:01 AM](#)

Hi! Tahnk you for sharing !
But it seems like you forgot to include "streaming.h" with library ? Is that intentional?

[Reply](#)



○
[Thijs Elenbaas](#)
[September 30, 2013 at 9:17 AM](#)

Hi, you're welcome! Indeed, I didn't include streaming.h. Did you get an error? The streaming library is not essential and the library should compile without it.

[Reply](#)

■
[xchg.ca](#)
[September 30, 2013 at 4:38 PM](#)

No, it doesnt compile. Had to find and add manyally streaming library to my sketch !

[Reply](#)

8.
[Nate](#)
[November 1, 2013 at 9:13 PM](#)

Hi, and thanks for this great library! I've got it working perfectly and it is exactly what I was looking for!

I have a question about binary send/receive of arguments: it appears there is no support currently for sending/receiving structs as binary. I think this is a extremely useful way to send data and would love to see it supported. Also, another question: are binary arguments escaped? How does that work if one of the bytes is the escape char? Are binary arguments still delimited by a comma? How does that work if one of the bytes is a comma?

Also, probably just an oversight, but there is no support on the Arduino side to read arguments as long. And there is also no way to read the ArgOk status after reading in an arg. I have rectified this for my own use, but I'd suggest making those changes for the next release.

Thanks again for a great library!

[Reply](#)



○
[Thijs Elenbaas](#)
[November 4, 2013 at 10:13 PM](#)

Hi.

Great to hear that you like it. And more has yet to come 😊

About the binary arguments. Indeed, there currently is no struct support. On the Arduino side it is quite easy to serialize a struct, but until recently I believed it was (almost) impossible to de-serialize the C-type structs. This is why I didn't include it. A weeks ago a friend pointed me to the interop functionality of C#, and I now believe I can get transferring structs to work. The upcoming update won't include this, probably, but one after that may.

All binary arguments are, and must be, escaped, otherwise the data will get garbled. If one of the bytes is an escape char, it will also be escaped. So a \ becomes a \\ . While programming I discovered this requires some bookkeeping, because you can not automatically assume that the character after a \ is automatically escaped, because this \ could be escaped as well. Still, I think I managed to keep the code that does this booking small and elegant.

Arguments are still delimited by -unescaped!- comma's. The arguments can consist of comma's, but must be escaped.

I will add the Long type to the next iteration of the library.

[Reply](#)

9.

Tomas

[November 10, 2013 at 2:27 AM](#)

Trying to run the examples. kStatus; and kSetLed, 1; works great.

But trying to kFloatAddition fails. How's the syntax for passing the float arguments to kFloatAddition? ATmega1280 board. Thank you.

[Reply](#)



o

[Thijs Elenbaas](#)

[April 6, 2014 at 4:34 PM](#)

Hi, How does it fail? The syntax is pretty simple, for the SendAndReceiveArguments sample I believe the command is: 2, float 1, float 2; . If you use the C# application, you could try to set the `SerialSettings.DtrEnable = true;`. Good luck

[Reply](#)

10.

Nate

[November 12, 2013 at 8:22 PM](#)

Hi again, and thanks for the helpful reply. I hope you're able to get support for structs figured out! I don't have much C# experience so I won't be of much help there. However, I've been doing some thinking about the support for binary arguments. I think if you take a slightly different approach, things might work more elegantly. Here is my suggestion:

- Make it so a command can have only binary arguments or ASCII arguments, not both. The entire command is sent as binary or ASCII.
- Send 0xFF as the first byte if the command will be binary, which would be followed by the command number (in binary), followed by the arguments, like so: 0xFF-CommandNum-Arg1ByteCount-Arg1Bytes-Arg2ByteCount-Arg2Bytes-...-ArgNByteCount-ArgNBytes.
- You could then do away with escaping for binary commands since the command is now sent as pure binary and not some combination of ASCII (for the argument delimiters) and binary.
- This format would also allow for additional error checking.
- ASCII commands/arguments could continue to work the same way.

It just seems like if you're going to support binary communication, it should be purely binary. And the point of sending as binary is generally speed and size, and eliminating escaping and argument delimiters will help with both (although admittedly only slightly).

Anyway, just a suggestion, take it or leave it. Looking forward to the next update!

[Reply](#)

11.

Russell Crow

[November 19, 2013 at 3:55 PM](#)

Thijs,

Your version of CmdMessenger rocks! It works really well! Easy to implement with solid results.

Thanks,
Russell

[Reply](#)



o

[Thijs Elenbaas](#)

[December 1, 2013 at 11:28 PM](#)

Ha! These are the responses that make my day 😊

[Reply](#)

12.

luke

[January 1, 2014 at 6:18 PM](#)

Great Work Thijs,

just one thing: Can you add an example of what I have to write in the the serial Comm window to test your examples and the expected output?

Thanks Luke

[Reply](#)



o

[Thijs Elenbaas](#)

[January 4, 2014 at 12:29 AM](#)

Hi Luke,

Thanks! I've added an example in v3.5 on how to communicate with CmdMessenger from the Serial Console. In this way you can use it as a kind of shell and interact with your Arduino. Let me know what you think.

Ps. I'm always interested in hearing what people are using my libraries for. 😊

Thijs

[Reply](#)

■

luke

[January 10, 2014 at 1:39 AM](#)

Hi Thijs, thank you for the example. Well my Projekt is a "polargraph". I will post some pictures as soon it works (if it works).

[Reply](#)



■
Thijs Elenbaas
[January 27, 2014 at 10:15 PM](#)

Great! I'm interested!

[Reply](#)

13.

Jonas Otto

[January 14, 2014 at 9:43 PM](#)

How can i include your library in c#? I always geht errors with the "using..."

[Reply](#)



○

Thijs Elenbaas
[April 7, 2014 at 8:58 AM](#)

This is too little information to give a good answer, but have you added the assembly to you solution?

[Reply](#)

■

Jonas Otto

[April 7, 2014 at 10:13 AM](#)

I solved it in my own

[Reply](#)

14.

Floyd

[January 26, 2014 at 9:26 PM](#)

Does the readStringArg work? Buffer was zero

I tried this and the "compare", and "copy", and did not get any results

Console data sent:

21,test;

```
// Ex:21
void kCMD_SET_CMD() // SET COMMAND
{
  if(cmdMessenger.available())
  {
    if(cmdMessenger.compareStringArg("test"))
      Serial.println("good");
    else
      Serial.println("bad");
  }
  cmdMessenger.sendCmd(kACK,"kCMD_SET_CMD");
}
```

Console Output:

Start Loop

3,Unknown command;1,Arduino ready;bad

1,kCMD_SET_CMD;

Do you have an examples?

[Reply](#)



○

Thijs Elenbaas
[January 27, 2014 at 10:07 PM](#)

Hmmm, I'll test this. To be honest, I haven't used strings that often.

[Reply](#)



15.

gwidenab

[March 3, 2014 at 3:49 AM](#)

Hi Thijs — thanks for the work on this functionality. Question: What do you advise for use from some other language, like Python, Java, etc? Is there some satisfactory robust way to access the PC-side CmdMessenger library, or does it really require re-implementing all or part of the PC side in the other language? Also, is there a spec somewhere for the allowed message formats “on the wire”, or if not, which code can they most easily be derived from? Thanks!

[Reply](#)



16.

gwideman

[March 3, 2014 at 3:50 AM](#)

Hahaha — typos in name — doh!

[Reply](#)



17.

gwideman

[March 4, 2014 at 9:46 AM](#)

Hmmm, I’m somewhat confused at this point. On March 3 I posted a question:

“What’s the recommended way to use CmdMessenger from another language (say Python or Java) on the PC side: Is there a way to take advantage of the PC-side CmdMessenger library, or does it make more sense to re-implement the PC-side code in the language of choice?”

But that message has disappeared, while the message I later posted commenting on my misspelling of my name is still here “awaiting moderation”. Anyhow, mostly I’m just interested in the question, so any answer appreciated!

[Reply](#)



o

[Thijs Elenbaas](#)

[April 4, 2014 at 11:56 PM](#)

Hi, sorry to have lost your question. Hmm, It is typically difficult to integrate a managed library, that is a C# library that uses the .NET run-time, in a non-managed language. I would re-implement it. There are code converters from c# to Java (for example <http://www.cs2j.com/>), and I would use this as a starting point. From there I would fix the code until the samples give the same results as the C# code. If you would make such a library, I would happily include it in a next release of the Package

[Reply](#)

18.

Tomas

[March 6, 2014 at 2:49 PM](#)

Hi,
very nice work with that Library indeed. I would like to know how well does that library works with WPF GUIs, windows, controlers and so on.
thanks.

[Reply](#)



o

[Thijs Elenbaas](#)

[April 4, 2014 at 11:41 PM](#)

Hi,
I did not test WPF, but the things I did to make it responsive using winforms also apply to WPF. Nevertheless, I would say that the best way of finding out, is trying it ☺

[Reply](#)

19.

Bo Nilsson

[March 7, 2014 at 2:24 PM](#)

Hi,
I have just found the cmdMessenger and try to get it to work on the Arduino. But I can't understand how it works.
I have load the Arduino with the receive sample and then.....
I use the serialmonitor in the pcprogram and try to send some textstring to the Arduino but I can't get the led to work.
Can you please help me to understand the cmdMessage library.
I also wonder if it work when I only send one letter. ex A.
No start or end character exists.
I know this is horrible but it is not me that have deside this.

[Reply](#)



o

[Thijs Elenbaas](#)

[April 6, 2014 at 1:56 PM](#)

Hi Bo,
I’m sorry, but this is really not enough information for me to help you out. Please read the blog posts, the explanation on the Arduino Playground, and then start from the simplest sample up to the more complex samples. Good luck!

[Reply](#)

20.

gstercken

[March 30, 2014 at 6:52 AM](#)

Hi Thijs,

first of all, thanks for the great work! I really like your implementation.

However, I am running into a strange issue (Arduino Uno R3 Windows 8 64-Bit) that occurs in my own application, but I can reproduce it with your samples as well (namely the DataLogging): At the default rate of 115200 on both ends, the communication becomes unreliable after a short time. The initial Arduino -> PC message (kAcknowledge, Arduino has stated) is received and displayed. But as soon as the PC sends kStartLogging, the Arduino will reply with kError – command without attached callback. This is very reproducible at all rates above 9600 bps. Only when I go as low as 9600 it suddenly works. You have any idea what the issue could be?

Thanks, Guido

[Reply](#)



o

[Thijs Elenbaas](#)

[April 4, 2014 at 11:23 PM](#)

Hi Guido,

That's weird! No, I have not seen this before, but I did only minor testing with Windows 8. I will do this for the next release. But if you could pinpoint it further, that would be great! Have you tried sending the same messages by hand over the serial terminal? Is the code running correctly for you at 115200 on Windows 7?

[Reply](#)

■ [Kevin B.](#)

[June 26, 2014 at 8:14 PM](#)

Hi,

I seem to be having a similar problem, but on Windows 7, and I don't receive the initial acknowledgement. What I do see is:

Sent > 2;
Received > 1,[0] Command without attached callback;
Arduino has experienced an error

The "[0]" I added to the sketch (cmdMessenger.CommandID()) to find out what the "command without a callback" was.

When I connect to the sketch via a serial terminal it works fine. Similarly if I run the DataLogging program at 9600.

This may not be related, but the SendandReceiveBinaryArguments example will sometimes hang during the binary section. This was easily reproducible, but then it started working...

Thank-you

[Reply](#)

■ [Kevin B.](#)

[June 26, 2014 at 8:22 PM](#)

Hi again,

I didn't notice John's fix until after I sent this. Sending the StartLogging command twice seems to have solved the problem but that does seem a bit strange, since other examples don't need it.

At the moment that should let me get started!

Thank-you

[Reply](#)



■

[Thijs Elenbaas](#)

[July 1, 2014 at 10:15 AM](#)

It seems to have to do with timing and the interaction of the foreground thread with the background thread. I will try to fix it in the next release, but it is proving to be a difficult problem!

21.

John

[April 6, 2014 at 11:11 PM](#)

Hi Thijs,

Your library seems exactly what I need to have. However, I have some problems with it. I am running on Windows 7, 64 bit.

This is the situation:

- I can send commands from the PC to Arduino.
- I can send acknowledge commands from Arduino to the PC.
- From the PC I can request data to be sent from the Arduino to the PC, and I receive this data as part of the request command.

So everything seems ok, but. I cannot get the PC to receive commands sent by the Arduino (initiated by the Arduino). The "Attached" method never gets called, although I can see that the data is sent by the Arduino.

Also the DataLogging example you have provided does not work. It is exactly the same, data is sent (checked that), but the attached callback method is never called.

I am using version 3.3 downloaded from your website (note that this is outdated on your website as I see that there is a 3.5 version available on GitHub).

Any idea what could be going wrong?

Any help is obviously appreciated.

Regards,

John.

[Reply](#)



o

[Thijs Elenbaas](#)
[April 7, 2014 at 8:56 AM](#)

Hi John,

I believe that I have fixed a bug that may solve this issue after the 3.3 release, so you could try the GitHub version. I am still in the progress of testing the next release, so use it at your own risk 😊. What you can also try, is to set the DtrEnable on the SerialSettings object. Some Arduino types and clones seem to need this. Also, you could try to see what happens if you reduce the transfer speed on both sides. Please let me know what you find!

Thijs.

[Reply](#)

■

[John](#)
[April 7, 2014 at 7:52 PM](#)

Hi Thijs,

I already tried both the DTREnable as well as the lower bitrate. Both does not have any impact.

I quickly need to get some simple HW going for which it is way too much work / expensive to create a dedicated board, I picked the Arduino to quickly get going, I hope your latest version will work. I will try this later.

Thanks,

John.

[Reply](#)

■

[John](#)
[April 7, 2014 at 8:04 PM](#)

Hi Thijs,

I just checked, the version from Github does not work either, same problem.

Any ideas left?

Regards,

John.

[Reply](#)



■

[Thijs Elenbaas](#)
[April 7, 2014 at 8:57 PM](#)

I'm baking a package of my work in progress version. Please try the samples that come with this version. It also comes with the beginnings of some unit testing, maybe you can try that. Also, what board are you using? I personally prefer the Arduino compatible Teensy 3.1. A lot faster, especially for serial over USB data transfer. Not a 100% compatible with all libraries, though.

■

[John](#)
[April 8, 2014 at 9:53 PM](#)

Hi Thijs,

I think that there is a race condition somewhere, your own data logging demo does not work out of the box for me.

When I duplicate the start logging command (send it twice) the demo works. The same fix also fixes my own project.

So for me it is ok now, thanks a lot for the library!

P.S. Can you remove my name and previous job environment from your posting?

Thanks,

John.

[Markus Baden](#)
[May 26, 2014 at 9:13 AM](#)

Thanks for your work on the library, I used version 2 of it in one of my Arduino projects [1] and recently updated to version 3.6.

Wow, what a difference in usability. Very impressed!

[1] <https://github.com/beercanlah/ardumashtun>

[Reply](#)



o [Thijs Elenbaas](#)
[July 1, 2014 at 10:51 AM](#)

Hey Thanks, that is a big compliment! I'm open to donations in the form of your home-brewn beer 🍺 Good luck with you project!

[Reply](#)

23.

[Thein Lin Aung](#)
[June 4, 2014 at 1:16 PM](#)

Hi John,

I have tested the library and it worked very well. I am happy with the results and glad to have discovered this great software.

I have one problem though, which is to make the software run in a VB.net program. I have some projects already written in VB and don't want to re-write in C# (I am not familiar with C#). So I'd be happy if there is an easy way to port the library to VB or compatible with VB projects.

[Reply](#)



o [Thijs Elenbaas](#)
[July 1, 2014 at 10:47 AM](#)

Hi, that is great to hear! You can simply reference and use the C# library in your VB programs. I may add VB versions of the examples in a later version.

[Reply](#)

■ [Thein Lin Aung](#)
[July 31, 2014 at 1:19 PM](#)

By reference library, do you mean by the
Dll file produced from the C# project?

[Reply](#)

■ [Thein Lin Aung](#)
[August 1, 2014 at 5:53 PM](#)

If so, how do you write the callback functions in VB and call `_cmdMessenger.Attach()`? Better yet, I'll very much appreciate if you could give a sample written in vb.
Thank you for your support.

[Reply](#)

24.

[Peter](#)
[July 1, 2014 at 6:57 AM](#)

Hi Thijs,

My first time using cmdMessenger, I opened your receive example in Arduino, and also opened the Arduino serial terminal, now I want to turn the led on, what command message I need to send? Another thing, does this package support hyperterminal communication?

Thanks

Peter

[Reply](#)



o [Thijs Elenbaas](#)
[July 1, 2014 at 10:17 AM](#)

You might want to take a look at ConsoleShell as an example to use from the terminal. The others are more focused on the communication between PC and Arduino.

[Reply](#)

■ [Peter](#)
[July 3, 2014 at 3:04 PM](#)

Thank you for your instruction,Thijs! I will try it, if later on anyproblem encountered I will consult you.

[Reply](#)

Leave a Reply

Required fields are marked *.

Name *

Email *

Website

Message *

☒ Notify me of follow-up comments via e-mail

[← Previous Post](#)

[Next Post →](#)

Meest recente berichten

- [Arduino to PC messaging – improved speed, compatibility and testing](#)
- [Webvidcoder – making video work in every browser and on every device](#)
- [Arduino to PC messaging – More devices, protocols, speeds](#)
- [Arduino-to-PC messaging](#)
- [Analyzing and optimizing performance of a WordPress website](#)

Categorieën

- [Arduino](#)
- [DCF77](#)
- [General](#)
- [Image processing](#)
- [Physics](#)
- [Synology](#)
- [Web development](#)

Copyright 2011 Thijs Elenbaas