Subject: Re: serial.dll for serial port under IDL
Posted by Wout De Nolf on Mon, 20 Oct 2008 15:21:01 GMT
View Forum Message <> Reply to Message

Is this what you're talking about?
http://imac-252a.stanford.edu/programs/IDL/idl_serial/

How do you know there's no reaction from the device on COMM_WRITE?
What errormessage is COMM_READ giving? Did you try calling it from a C
(or C++) program?

If you have a compiler, start debugging serial.c by attaching to a
running idl process (IDL7.0: C:\Program
Files\ITT\IDL70\bin\bin.x86\idl_opserver.exe, IDL6.x: if I remember
correctly this was idlde.exe...) and then issue the COMM_READ in IDL
and set a breakpoint in the c-code.

On Sat, 18 Oct 2008 09:50:10 -0700 (PDT), thejll@mail.dk wrote:

> I am unable to get the serial.dll and its dlm to work. Can anybody
> show me a piece of IDL code that uses the serial.dll and dlm to read
> from or write to the serial port?
>
> I'd like to be able to do things like read from a GPS or a telescope
> (whic accepts strings printed to the serial port).
>
> As seen on this group in about 2004 the dll and the dlm has been
> discussed - but no use was shown. I too can use the COMM_OPEN and
> COMM_CLOSE calls. I can get the COMM_WRITE to report the number of
> characters written but there seems to be no reaction from the device
> written to and COMM_READ never reports anything but failure.
>
> It may be significant to say that I use IDL 6.3 on a laptop and use a
> serial-USB dongle. Yes, the dongle works - e.g with a GPS and
> dedicated GPS software.
>
> Can anyone suggest something that works?
>
> Peter Thejll
> Denmark

Subject: Re: serial.dll for serial port under IDL
Posted by Rick Towler on Mon, 20 Oct 2008 18:19:43 GMT
View Forum Message <> Reply to Message

Are you sure the serial dlm isn't working? Isolate the issue by doing a

loopback test. Simply short pins 2 and 3 of the DB9 connector as
described here:

http://zone.ni.com/devzone/cda/tut/p/id/3450

Then write a simple program in IDL that sends some data then reads it.
Once you can send data to yourself then connect up the external devices
and try to communicate with them.

If you still can't get it to work you can try recompiling it. I know
that this code has been kicking around for a while so maybe the binary
isn't compatible with 6.3?

-Rick

thejll@mail.dk wrote:
> I am unable to get the serial.dll and its dlm to work. Can anybody
> show me a piece of IDL code that uses the serial.dll and dlm to read
> from or write to the serial port?
>
> I'd like to be able to do things like read from a GPS or a telescope
> (whic accepts strings printed to the serial port).
>
> As seen on this group in about 2004 the dll and the dlm has been
> discussed - but no use was shown. I too can use the COMM_OPEN and
> COMM_CLOSE calls. I can get the COMM_WRITE to report the number of
> characters written but there seems to be no reaction from the device
> written to and COMM_READ never reports anything but failure.
>
> It may be significant to say that I use IDL 6.3 on a laptop and use a
> serial-USB dongle. Yes, the dongle works - e.g with a GPS and
> dedicated GPS software.
>
> Can anyone suggest something that works?
>
> Peter Thejll
> Denmark

Subject: Re: serial.dll for serial port under IDL
Posted by peter.thejll on Wed, 22 Oct 2008 13:05:18 GMT

View Forum Message <> Reply to Message

Hi, and thanks for your reply.

On Oct 20, 5:21 pm, Wox <s...@nomail.com> wrote:

> Is this what you're talking about?http://imac-252a.stanford.edu/programs/IDL/idl_serial/
>

Yes - these files can also be found on the IDL site as a 'plugin'.
They exist with slightly different sizes and dates so I figure they
have been updated by some along the way. I downloaded a set from 2007.

> How do you know there's no reaction from the device on COMM_WRITE?

COMM_WRITE answers with the number of characters sent, so if I send a
3-character string the value of h in h=COMM_WRITE(...) is 3. I
therefore think the WRITE part is OK - but when I read I get an error
message along the lines 'failed to read' (I am not at the computer
with the software on right now).

> Did you try calling it from a C
> (or C++) program?

I am able to read the device on the serial port with a Visual Basic
programme. It basically prints strings to the port or reads the port.
No problem.

>
> If you have a compiler, start debugging serial.c by attaching to a
> running idl process (IDL7.0: C:\Program
> Files\ITT\IDL70\bin\bin.x86\idl_opserver.exe, IDL6.x: if I remember
> correctly this was idlde.exe...) and then issue the COMM_READ in IDL
> and set a breakpoint in the c-code.
>

Mmmmm. This is beyond me at the moment! Thanks for the suggestion,
though!

I am catching your drift, namely 'how do I know the port is not being
read'? I will worry about that one. Thanks for now!

Peter

Subject: Re: serial.dll for serial port under IDL
Posted by peter.thejll on Wed, 22 Oct 2008 13:07:34 GMT

View Forum Message <> Reply to Message

Hi, Thanks for your suggestions.

On Oct 20, 8:19 pm, Rick Towler <rick.tow...@nomail.noaa.gov> wrote:

> Are you sure the serial dlm isn't working?

No.

> Isolate the issue by doing a
> loopback test. Simply short pins 2 and 3 of the DB9 connector as
> described here:
>
> http://zone.ni.com/devzone/cda/tut/p/id/3450
>
> Then write a simple program in IDL that sends some data then reads it.
> Once you can send data to yourself then connect up the external devices
> and try to communicate with them.

Good idea! I am not much for soldering though - is the cable you
describe a 'null-modem cable'? I have one of those.

Cheers,

Peter

Subject: Re: serial.dll for serial port under IDL
Posted by Rick Towler on Wed, 22 Oct 2008 16:10:46 GMT

View Forum Message <> Reply to Message

peter.thejll@gmail.com wrote:
> Hi, Thanks for your suggestions.
>
> On Oct 20, 8:19 pm, Rick Towler <rick.tow...@nomail.noaa.gov> wrote:
>> Are you sure the serial dlm isn't working?
>
> No.
>
>> Isolate the issue by doing a
>> loopback test. Simply short pins 2 and 3 of the DB9 connector as
>> described here:
>>
>> http://zone.ni.com/devzone/cda/tut/p/id/3450
>>
>> Then write a simple program in IDL that sends some data then reads it.
>> Once you can send data to yourself then connect up the external devices
>> and try to communicate with them.
>
> Good idea! I am not much for soldering though - is the cable you
> describe a 'null-modem cable'? I have one of those.

No soldering required. You can use a paper clip :). That null modem cable should have female connectors at both ends so you can plug one end into your computer then short pins 2-3 at the other end of the cable.

Use Hyperterm to test your cable as described on that same webpage I linked above. Once you know your cable is working then you can try IDL. The test in IDL will be as simple as this:

```
IDL>  h=comm_open('COM1',BAUD=9600,DATA=8,MODE=3,PARITY='N',STOP=1 )
IDL> b=comm_write(h,'This is a test')
IDL> data=comm_read(h)
IDL> print, data
   84 104 105 115  32 105 115  32  97  32 116 101 115 116
IDL> print, string(data)
This is a test
```

It looks as if the original .dll was built for IDL 5.6. I just recompiled it for 6.4 for the test above. The issue might simply be that you need to recompile. I know you're working with 6.3, but I'll send you the .dll I just compiled and you can try that.

-Rick

Subject: Re: serial.dll for serial port under IDL
Posted by Rick Towler on Fri, 24 Oct 2008 16:10:28 GMT

View Forum Message <> Reply to Message

Peter sent some test code to me and I was able to reproduce the problem. It turns out that the issue is that IDL is "too fast" in that if you write and then immediately read, IDL polls the serial port before anything is in the serial port buffer. For small amounts of data this means that you'll see nothing as Peter did. For large amounts of data it means that you will not see all of the data. For example, sending 4000 bytes of data and immediately reading results in (on my machine) only 3984 bytes being returned. If you insert a pause between the write and the read (wait, 0.1) you will Rx all 4000 bytes.

The issue is a bit more complicated than I have presented since you never get those last few bytes even if you call comm_read again. In my case I should have 16 more bytes lying around but subsequent calls yield nothing. So the question is, is this a result of a freak timing issue that only comes up when you're looping back? Or is there a bug in the .dlm?

I'm leaning toward the prior but I don't know. I do know that if you are providing you're own buffer to comm_read it better be the same size or smaller than the serial port buffer or you get some weird results.

Post your results Peter.

-Rick

```
; test program for serial port reader
; NOTE: you MUST set the COM port number correctly first time or code
crashes
;
portstring='COM3'
print,'Trying to open ',portstring
 handle=comm_open(portstring,DATA=8,BAUD=4800,STOP=1,PARITY=' N',mode=3)
print,'handle from OPEN=',handle
;--------- WRITE
data=byte('6b')
h=comm_write(handle,data)
print,'Result of write : ',h
;--------- READ
buf = bytarr(5)
h=comm_read(handle,BUFFER=buf)
print,'Result of read : ',h
print,'Buffer read: ',buf
;--------- CLOSE -------------
res=comm_close(/all)
print,'result of comm_close command:',res
end
```

Rick Towler wrote:
>
>
> peter.thejll@gmail.com wrote:
>> Hi, Thanks for your suggestions.
>>
>> On Oct 20, 8:19 pm, Rick Towler <rick.tow...@nomail.noaa.gov> wrote:
>>> Are you sure the serial dlm isn't working?
>>
>> No.
>>
>>> Isolate the issue by doing a
>>> loopback test. Simply short pins 2 and 3 of the DB9 connector as
>>> described here:

>>>
>>> http://zone.ni.com/devzone/cda/tut/p/id/3450
>>>
>>> Then write a simple program in IDL that sends some data then reads it.
>>> Once you can send data to yourself then connect up the external devices
>>> and try to communicate with them.
>>
>> Good idea! I am not much for soldering though - is the cable you
>> describe a 'null-modem cable'? I have one of those.
>
> No soldering required. You can use a paper clip :). That null modem
> cable should have female connectors at both ends so you can plug one end
> into your computer then short pins 2-3 at the other end of the cable.
>
> Use Hyperterm to test your cable as described on that same webpage I
> linked above. Once you know your cable is working then you can try IDL.
>   The test in IDL will be as simple as this:
>
> IDL> h=comm_open('COM1',BAUD=9600,DATA=8,MODE=3,PARITY='N',STOP=1)
> IDL> b=comm_write(h,'This is a test')
> IDL> data=comm_read(h)
> IDL> print, data
>   84 104 105 115 32 105 115 32 97 32 116 101 115 116
> IDL> print, string(data)
> This is a test
>
>
> It looks as if the original .dll was built for IDL 5.6. I just
> recompiled it for 6.4 for the test above. The issue might simply be
> that you need to recompile. I know you're working with 6.3, but I'll
> send you the .dll I just compiled and you can try that.
>
> -Rick