
Subject: serial.dll for serial port under IDL
Posted by [thejll](#) on Sat, 18 Oct 2008 16:50:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

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 [thejll](#) on Fri, 24 Oct 2008 19:11:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Rick!

I can now report success. The trick was, as Rick says, in the waiting - IDL is indeed reading the port too quickly after a write. This must be a problem for my particular device (a Meade ETX90 telescope that takes commands over the serial port). By testing the code below and inserting various waits it became clear that my device often wants to wait more than 0.05 seconds, and can wait as much as 1 second after sending a command to the port before reading the response.

Here is the working code:

-----start code insert

```
portstring='COM3'  
print,'Trying to open ',portstring  
h = comm_open(portstring, DATA=8,BAUD=9600, STOP=1, $  
  PARITY='N', MODE=3);, BUFFERSIZE=8192)
```

```

print,'handle from OPEN=',h

;----- WRITE
data = bytarr(19) + byte(':Gt#')      ; this command reads the
telescope                               ; latitude from the
Autostar (LX200 command set)
b = comm_write(h, data)
print,'Result of write : ',b

;----- READ
wait, .15
data2 = bytarr(19)
n=0
while (n eq 0) do begin
  b= comm_read(h, buffer=data2)
  print,'Result of read in loop : ',data2
  x=where(data2 gt 0, n)
endwhile
print, 'Result of read once n was not 0: ',data2(0:n-1)

;----- CLOSE -----
res=comm_close(/all)
print,'result of comm_close command:',res
end

----- end of insrted code -----

```

Example output of the above looks like this:

```

IDL> .GO
Trying to open COM3
BAUD=9600 DATA=8 PARITY=N STOP=1
handle from OPEN=      484
Result of write :      4
Result of read in loop : 43 53 53 223 52 49 58 48 48 35
0 0 0 0 0 0 0 0 0
Result of read once n was not 0: 43 53 53 223 52 49 58 48 48
35
result of comm_close command:      1

```

-----end example output

Thanks a lot for the help!

Peter Thejll

On Oct 24, 6:10 pm, Rick Towler <rick.tow...@nomail.noaa.gov> wrote:

```
> 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.the...@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- Hide quoted text -
>
> - Show quoted text -

Subject: Re: serial.dll for serial port under IDL
Posted by [fvelarde6](#) on Sun, 15 Jul 2012 01:39:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Saturday, October 18, 2008 12:50:10 PM UTC-4, (unknown) 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

Hi guys, I was serarching the serial.dll but I could not found it. Please, could someone send me the serial.dll? I will be very grateful.

Regards
Fer.
