

Re: Trying to send RS232 with PIC sends incorrect data in TeraTerm

# Re: Trying to send RS232 with PIC sends incorrect data in TeraTerm

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-04/msg02355.html>

---

- *From:* [jbitz34@xxxxxxxx](mailto:jbitz34@xxxxxxxx)
  - *Date:* 11 Apr 2007 22:27:57 -0700
- 

On Apr 11, 4:26 pm, "Roger Hamlett"  
<rogerspamigno...@xxxxxxxxxxxxxxxxxxxx> wrote:

<jbit...@xxxxxxxx> wrote in message

[news:1176324076.120588.120280@xx](mailto:news:1176324076.120588.120280@xx)

On Apr 11, 1:49 pm, "Roger Hamlett"  
<rogerspamigno...@xxxxxxxxxxxxxxxxxxxx> wrote:

<jbit...@xxxxxxxx> wrote in message

[news:1176252759.067057.96530@xx](mailto:news:1176252759.067057.96530@xx)>

So we  
shortened the message to just output "A".

The display on the Tera Term is now "\_"  
(underscore)

The binary for A is 1000001  
The binary for \_ is 1011111.

Just an update.

Ok.  
Now multiple things. If you have download rights for the  
compiler,

Re: Trying to send RS232 with PIC sends incorrect data in TeraTerm

3.249,  
is available to download. Install this into another directory,  
and see  
if  
the behaviour is the same. If so, you have at least ruled this  
out.  
Now, remember that for asynchronous serial, the data is sent  
LSBit  
first.  
Also, the 'start', requires a '0' (low) at the input, before the  
data  
byte  
itself is sampled. Then there is a high 'stop bit' at the end. So,  
'A'  
is:

1010000010x1, with '0' being 0v, and '1' being 5v (the 'x' is  
the  
parity  
bit, which would depend on whether this is enabled, and how  
it is set)

The \_, is:

101111010x1

Now the chip will start reading, when it sees a 0v for the first  
time.  
Notice the five successive '1's in the second stream, and the  
five  
successive '0's in the first.  
If you invert the bit pattern for the 'A', you get:

010111101/x0

Notice how if you shift this left one bit, it closely resembles  
what  
you  
are receiving.

Re: Trying to send RS232 with PIC sends incorrect data in TeraTerm

It looks suspiciously like you might be lacking a signal inversion in the connection to the PIC (RS232 uses  $-3$  to  $-25v$  for a '1', and  $+3$  to  $+25v$  for a '0'). Are you sure the converter you have is USB to RS232?. You speak about it as being expensive, but these are incredibly cheap (well under 10\$). I have a nasty suspicion, that your converter, might be something like USB to RS485, rather than RS232!. This would explain its price. While I agree, that the USB-RS232 converters can sometimes be troublesome, most now work OK. I would try the simple test of connecting it to another PC with RS232 (can you borrow one from a friend?), using just the minimum 3 wire connection (TX to RX, RX to TX, and GND to GND). At least this will allow you to 'rule out' the unit if it works. I really would be looking at the snalling polarity, and the connections round the MAX232.

Best Wishes

Well I'm a poor college student, so 20 dollars is expensive to me ;-)

The datasheet for the part indicates USB to RS232 functionality; and the chip that resides on the breakout board is the cp2102 which lists as its applications RS232 to USB on its datasheet.

Hint. Common mistake.

Same one in CCS.

They have their setup line called '#useRS232', but in fact it sets the chip up to produce TTL asynch serial, which requires the MAX232 (or

Re: Trying to send RS232 with PIC sends incorrect data in TeraTerm

similar), to 'generate'RS232signalling. In fact the same software setup is used with different hardware, to provide RS485, or a number of other similar standards.

The CP2102, is exactly the same. It does not provide USB toRS232 functionality, it provides USB to asynch TTL serial, and also requires a MAX232 or similar transceiver to generate 'RS232'. If you look at the data sheet for the chip, it shows an 'externalRS232transceiver' after the chip, to generateRS232signalling levels, and signal polarity.

Now you are talking about a 'breakout board', which suggests you are accessing the CP2102 directly. If so, it's connections, want to go directly to thePIC, without the MAX232 transceiver. This would explain the signals being inverted, since each MAX232 inverts the signals, as well as shifting the voltage levels used, and you have one missing. You need to either run with none (TTL signalling), or with two -RS232signalling between these. Check the voltages being produced from the 'breakout box', with the scope. I suspect you are going to find these are basically 5v levels. If so, get rid of the MAX232. I just hope you have not damaged the chip, since the MAX232, delivers +/-12v signalling, and if I am right, you have been feeding this into a chip rated for 5v signalling...

Here is the sparkfun product page:

[http://www.sparkfun.com/commerce/product\\_info.php?products\\_id=198](http://www.sparkfun.com/commerce/product_info.php?products_id=198)

Note that this correctly says this is a USB to serial adapter, not a USB toRS232adapter. Get rid of the MAX232 on thePIC.

I am not sure what snalling polarity is. Our connections to the Max232 have been checked by many people, and we have done so according to the numerous schematics we have found. Also, we know that the Max works at least for data sent out from the PC and shorted back 'round through the pins.

The problem is that you don't need the MAX232, since you are not usingRS232signalling anywhere...

Best Wishes

I will definitely check the voltages coming out of the Tx pin on the USB to serial board. We actually ruined a PIC and we had assumed it was because the Tx pin was delivering 12V, and thus added the Max; but I guess it was some other mistake. I will let you know what I find.

The chips have worked fine when shorting the lines, as late as today, so I hope that means that the incorrect delivery voltage has not hurt it.

Re: Trying to send RS232 with PIC sends incorrect data in TeraTerm

Re: Trying to send RS232 with PIC sends incorrect data in TeraTerm

I guess the old saying, about what happens when you assume, is true.  
I had assumed that since it was creating 9 pin serial it was creating  
RS232 level signals. It will be interesting to see if removing the  
middle man is the solution we were hoping for.