Graphical user interface

Description automatically generated

1 bit time = 104 uS, 9615 BAUD

STOP

START

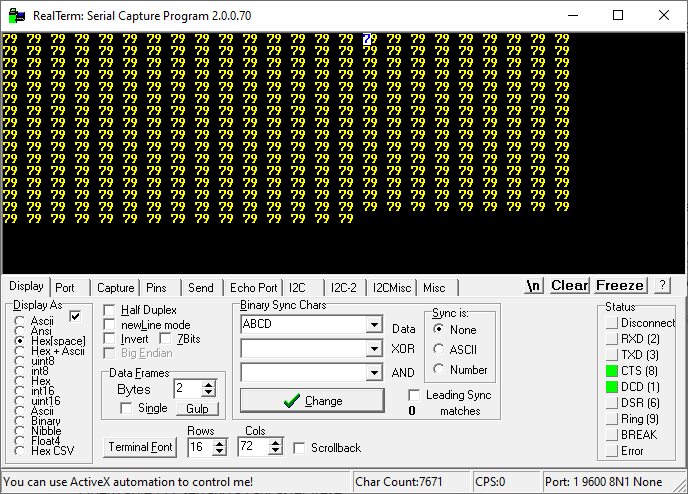
1 1 0 0 0 0 1 0

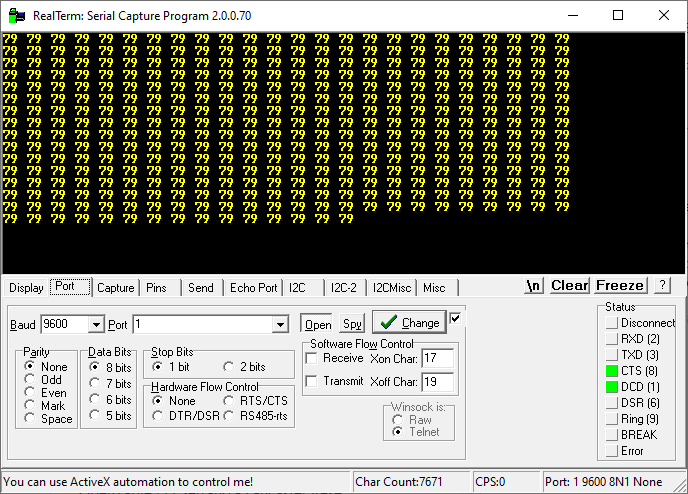
This is the Tx output from the B serial port on a TMS320F28069M microcontroller.

Set up as 9600 – 8,None,1, None.

The order of the bits as read should be (from right to left on the scope screen) 0100 0011, or 43 hex, corresponding to ASCII “C”.

Here’s the RealTerm output (same results for TeraTerm) for the above input - 2 screenshots to show “Display” tab and “Port” tab:





There’s no simple transform to make me say “Oh, the bit stream is just getting inverted and then reversed”; or “the LSB and the MSB have been switched”; or “the whole thing got shifted by X places.” I can come up with similar machinations, but they only work for limited cases.

One thing that’s consistent is that if the actual ASCII representation of a character has 6 zeroes and 2 ones, the RealTerm readout will have 6 ones and 2 zeroes (which is what originally made me think things were getting inverted and shifted).

Here are some examples of other characters:

Letter “B” -

ASCII (and scope readout): 0100 0010

RealTerm Output: 0111 1011

If you take the ASCII representation and Invert, then reverse the order, then move the MSB into the LSB position, voila, you have the RealTerm readout.

Letter “A” –

ASCII (and scope readout): 0100 0001

RealTerm Output: 0111 1101

For this one, all you have to do is invert and flip the order.

There are more cases in which the techniques employed for the letters A and B work, but nothing consistent. No simple mapping appears to work at all for, amongst others, the letter “C” (the first example with the screenshots, above).