```
// FILE: serialPc-M-1_0b.sce - Works!  Fixed 1_0a
// DATE: 04/18/20 07:40
// AUTH: P.Oh
// DESC: PC USB RS485 connect to (Master) NXT.  Scilab running on PC sends
//       serial message e.g. " @90, -90" (without quotes) containing desired
//       angles for XL-320 Lego-based 2-DOF planar manipulator.  Master
//       (running PC-M-S-1_0a.nxc) processes this message (and sends to Slave
//       via Bluetooth).
// VERS: 1_0a: based on scilabPcSerialToNxt0_1h.sce
//       1_0b: Different angles for homework
// REFS: Works with Master NXT running PC-M-S-1_0a.nxc and Slave running
//       btS-R-1_0a.nxc

h = openserial(10,"4800,n,8,1"); // initialize PC's serial port
strHeader = " @"; // white space + at character
stringRoger = "ROGER";
stringRogerFound = 1; // not TRUE

for i = 1:4 // four different angle pairs
    // 1_0a used: (0,90); (90,0); (0,-90); (90,90) - and worked!
    // 1_0b: each coordinate must be sign followed by 3 characters
    //       here, created the string directly with specific format
    //       i.e. a 4 character string: sign and 3 characters
    //       the 3 characters are the digits or when value < 3, then
    //       substitute with white space
    if i == 1 then
        strPosition01 = "+102";  // NB1: sign and 3 digits
        strPosition02 = "+102";
    end
    if i == 2 then
        strPosition01 = "+90 "; // NB2: sign and 2 digits + white space = 3
        strPosition02 = "-90 "; // NB2: sign and 2 digits + white space = 3
    end
    if i == 3 then
        strPosition01 = "-90 "; // same as above
        strPosition02 = "+90 ";
    end
    if i == 4 then
        strPosition01 = "+45 ";
        strPosition02 = "+45 ";
    end

    strI = strcat([strHeader, strPosition01, ", ", strPosition02]);
    disp(strI);
    writeserial(h, strI); // transmit serially to Master NXT

    buf = readserial(h);
    // Check if Master ready to receive next string
    stringRogerFound = strcmp(stringRoger, buf); // 0: means identical strings
    while (stringRogerFound ~=0) // then NXT -> PC string not ROGER, so wait
       buf = readserial(h);
       stringRogerFound = strcmp(stringRoger, buf);
       sleep(200); // min about 50 ms before reading serial port again
    end; // exit reading serial port when ROGER received
    disp(buf);
    sleep(5000); // just slows down loop so user can see what's happening
end

disp("All done!");
closeserial(h)
```