Using RS232 serial link with SERVOPAC TTxxx drive

1) Introduction

The RS232 serial link can operate in BINARY or ASCII mode. The operation mode can be selected by object 0x2301,0. The communication BaudRate is defined by the object 0x2300,0.

GemDriveStudio software is communicating only with the BINARY mode.

Factory adjustements for the serial link are : BINARY mode and BaudRate = 19200.

Parameters 0x2300,0 and 0x2301,0 are both saved into the drive (DRIVEPAR.TXT file).

When the parameter 0x2300,0 = 0, the communication speed can be changed in the « Communication configuration » window of GemDriveStudio (drive in auto-baud mode).
For the ASCII mode of operation, both the ASCII mode configuration and the BaudRate must be saved into the drive. The value 0 for the parameter 0x2300,0 must not be set in ASCII mode (drive auto-baud not operating). Doing so, the communication will be out of order at the next drive power up. In this case CAN bus acces must be used to retore the serial link operation.

2) Application example

Multiaxis operation with both "BINARY" mode for GemDriveStudio and "ASCII mode" for host controller:
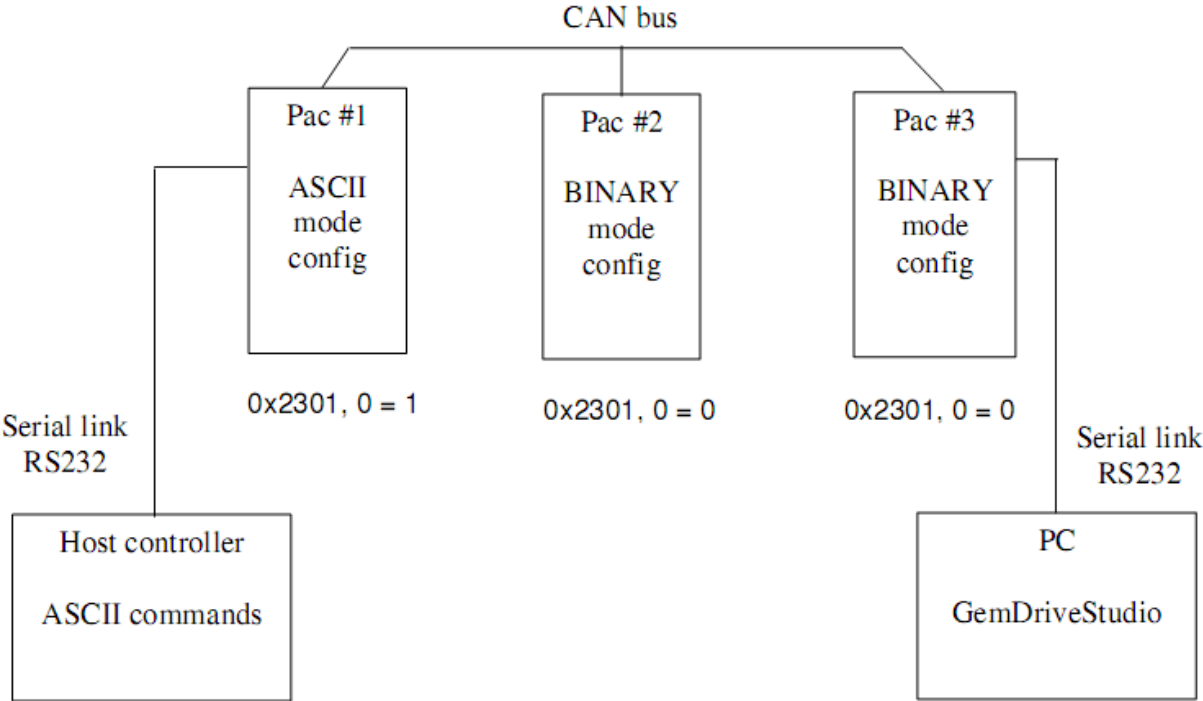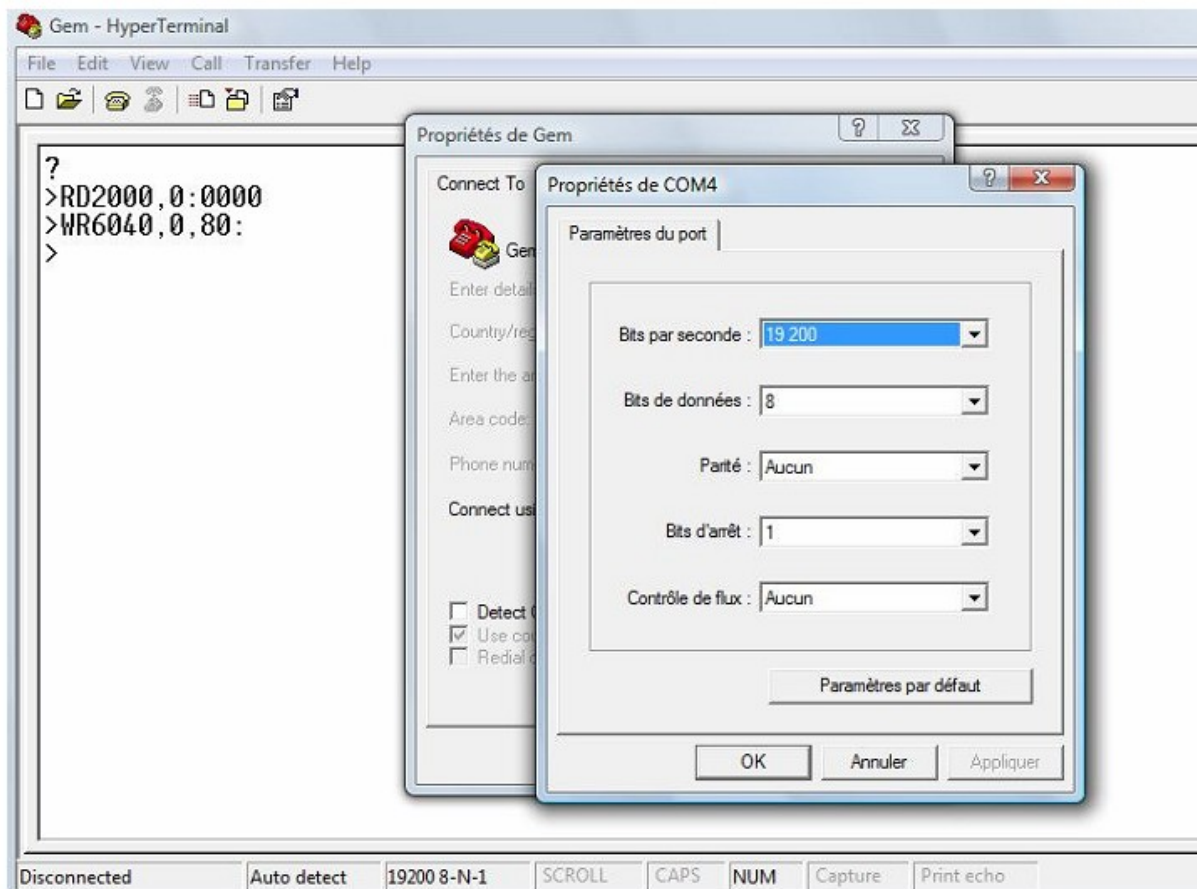
3) User guide

To change from "BINARY" mode to "ASCII mode", you need to write 0x2301, 0 = 1 by using the GDS dialog window. Then you can test the ASCII mode using the HyperTerminal from windows.

There are only 3 commands that you can use :

RDindex,sub	read an object
WRindex,sub,value	write into an object
RNaddress	change to a remote node

After change to a remote node, RD and WR are applied to this remote node.

Remark1: RD / WR a string is supported only for the local node (the axis with the serial link), not by a remote node (axes acessed via CAN bus).

Remark2: For connecting to the local node (the axis with the serial link), the command RN0 is equivalent to RNlocal_node_address.

4) TTxxx ASCII protocol

The serial link communication specifications for TTxxx amplifiers are listed below:
  8 Data bits,
  1 Stop bit,
  No parity,
  19200 Bauds.

Each instruction is coded as 2 ASCII characters with or without parameter.

Each instruction (which can be followed by an hexadecimal 16 bit or 32 bit parameter) sent to the amplifier must end with a "carriage return" character (ASCII code 13).

All these characters, except for the "carriage return", will be sent back by the amplifier (echo). The amplifier answer starts with a separation character ":" (ASCII code 58) possibly followed by an hexadecimal 16 bit or 32 bit value. The amplifier will then send a "carriage return", a "line feed" and ">".

Notes :
- If the amplifier does not know the instruction, it will send back "?" instead of ":".
- The parameters are always in hexadecimal.

Dialogue examples :

The PC sends the RN instruction:
  RN3
and a "carriage return" character (ASCII code 13) for ending the instruction.

The amplifier will answer with:
  RN3:
  >

":" indicates that the instruction has actually been decoded. The value 3 is stored in the corresponding variable. After the character "carriage return", the amplifier will also send the ">" character in order to indicate that it has taken the instruction into account.

If the PC sends the instruction:
  RN

The amplifier will answer with:
  RN:0003
  >

As there is no parameter, the amplifier sends back the parameter value (in this case: 3) after the characters RN and ":".