Annex A Transmissions with CRC32

To guarantee a secure data transfer also with a RS232 transmission and interfering field environment, the protocol offers a CRC32 procedure.

This support does not apply to the file transfers (for this separated block check numbers are used. See $FT File Transfer)

If a transmission with CRC32 should be carried out is always initiated by the host prior to every transmission.
For the procedure of a CRC32 transmission the following commands and parameter transfers are important:

ParameterTransfer: NextCRC32 (Implementation Host<->LJ3)
^0=NR Par1 = CRC32 Checksum

ParameterTransfer: CRCFailed (Implementation Host<-LJ3)
^0=FC Par1 = CRC32 Checksum

Command:: CRCok (Implementation Host<-LJ3)
^0!OK

Description of operation:
If you would like to have a secured transmission, the host calculates previous to every transmission the CRC32 checksum of the next transmission string and transfers it with ^0=NC to the printer.
It memorizes this CRC32 checksum, calculates and compares the CRC32 checksum of the transfered string for the next incoming communciation string.
If both are unequally a response with the checksum which has been calculated in the LJ3 will be sent to the host ^0=FC.
The host is then able e.g. to carry out a transmission repetition.
If the checksum is okay the printer responds (prior to the transmission of possible data) with ^0!OK.
If an inquiry of data (^0?XX)with CRC32 has been sent to the printer a CRC32 checksum is transfered from the printer to the host prior to every response line.

Example of a (Hyperterminal) sequence with CRC32: Query of the current used jobs (H = Host, L = printer):

^0=NR3957421711	➔ CRC32 of the following ^0?JL inquiry (H->L)
^0?JL	➔ inquiry itself (H->L)
^0!OK	➔ the confirmation of the LJ3 the CRC OK (H<-L)

^0=NR3560773416 ➔ CRC32 of the pending response (H<-L)

^0=JL\FFSDISK\JOBS\Testprint.job ➔ response itself (H<-L)