Unofficial Documentation of the

Behringer X32 Expansion-Slot Protocol

v1.1, 21.08.2025

More Information about a DIY card using these commands:

https://www.github.com/xn--nding-jua/xfbape

Inhalt

1. X-LIVE

1.1. Playback Commands

These commands are the most useful for a DIY card. They can be used to control MP3-player or other functions using the GUI of the X32-hardware (console) or the X32-Edit.

1.1.1. Select session

Command: *9B SESSIONNAME #Q#

Example: *9B582100A0#Q#

Answer: *9B000 MARKERCOUNT CHANNELCOUNT 0 NUMBEROFTOTALSAMPLES 0 ???

Example: *9B0000232000F4E900000000000#

1.1.2. Play session

Command: *9D#

Answer: *9D00#

1.1.3. Pause session

Command: *9E#

Answer: *9E00#

1.1.4. Stop session

Command: *9F#

Answer: *9F00#

If recording has stopped, card answers with a bunch of additional information using *9N-commands to update the GUI.

1.1.5. Seek session

Command: *9M SAMPLEINDEX #Q#

Example: *9M0009EDC0#Q#

Answer: *9M00#

1.1.6. Request first entry of TOC

Command: *9AF#

Answer: *9ASF TITLE #

Example: *9ASF582100A0#

1.1.7. Request additional TOC-entries

Command: *9AN#

Answer: *9ASF TITLE #

Example: *9ASF582100A0#

If there are no more entries, card sends a *9AEN#

1.1.8. Sending current sampleindex while playing

Answer: *9N22 SAMPLEINDEX#

Example: *9N22000004BF#

Sends the current playback-position as sampleindex to the X32

1.1.9. Sending "alive" command

Answer: *8BE#

This message has to be sent every 5 seconds

1.2. Vital Commands

Vital Commands have to be implemented to identify the installed cards and cause the X32 console to load the desired user-interface (especially the X-LIVE-controls). Most of the commands are called only once.

1.2.1. Identification of card

Command: *8I#   **and**   *8R#

Answer: *8 NAME : FIRMWAREVERSION #

Example: *8X-UREC:A:12#

Up to now, the following card-strings are tested:

- *8X-UREC:A:12#
- *8DN32-UREC:A:12#
- *8X-DANTE:A:1#
- *8X-MADI:A:2#

Based on screenshots of setup-screens, the following strings should work, too:

- *8X-USB:A:2#
- *8X-UF:A:1#
- *8X-ADAT:A:2#

1.2.2. Request size of card

Command: *9N#

Answer: *9N CARDNUMBER 0 CARDSIZE USEDSPACE #

Example: *9N1001DB830000200000#

Caution: the calculation is not correct at the moment. Card-Size is shown correctly, but used space to calculate remaining time is not correct.

1.2.3. Request total size(?)

Command: *9G#

Answer: *9G CARDNUMBER 0 CARDSIZE1+CARDSIZE2 #

Example: *9G1003B70600#

Its not clear, why we have to transmit the total size here.

1.2.4. Request markers

Command: *9C MARKERINDEX#

Example: *9C02#

Answer: *9C00 MARKERINDEX TIMEINDEX#

Example: *9C0002000004BF#

Important: with markerindex = 0 the total length of the current session is transmitted:

Example: *9C00#

Answer: *9C0000 TOTALSAMPLES#

1.3. Additional Commands

Additional Commands are "nice to have" and allow additional communication between X32 and a DIY card. But some of the have no special usage for a DIY card at the moment.

1.3.1. Switch between USB and SD-Card

Command: *8C8 OPTION DEVICE#

Example: *8C81C#

Answer: *8Y#

Option contains number between 0 and 5, to select number of channels to record

Device can be "C" for Card and "U" for USB

1.3.2. Detecting a new SD-card

Answer: *9N CARDINDEX 0 00007F40 00000000#

Example: *9N1000007F4000000000#

At the moment it is unclear, why 7F40 is sent. It represents 32576 in decimal and has no direct meaning on the first view.

1.3.3. Delete Session

Command: *9X SESSIONNAME #

Example: *9X582100A0#

Answer: *9Y00#

1.3.4. Start recoding new session

Command: *9H SESSIONNAME #

Example: *9H582100A0#

Answer: *9Y00#

1.3.5. Format SD-Card

Command: *9Q~#

Answer: *9Y00#*9N0700000000##

1.3.6. Create new marker

Command: *9I SAMPLEINDEX #

Example: *9I000004BF#

Answer: *9Y00#

1.3.7. Select SD-Card

Command: *9R CARD #

Example: *9R0#   or   *9R1#

Answer: *9R00#

1.3.8. Initialization-Commands

Command: *_G 5-DIGIT NUMBERS#

Example: *3G70000#

Answer: no

2. Other X-Cards

Beside the X-LIVE (or DN32-LIVE) there are several other cards that can be used. Most of the cards have limited functions compared to the X-LIVE card: X-UF, X-USB and X-ADAT have no special functions beside the conversion of 4x 8 channel TDM-streams to USB or ADAT.

Only X-MADI and X-DANTE cards have some special features, that could be interesting: X-MADI-cards can be used as clock-master, hence the samplerate of the console can be clocked by a DIY-card.

2.1. X-ADAT

2.1.1. Enable/Disable external Word-Clock

Command: *8C4Y#

Anwer: *8Y#

Command: *8C4N#

Anwer: *8Y#

State of Word-Clock can be checked using *8S# command

2.1.2. Read State-Command

Command: *8S#

Answer: *8YN#

First "Y" is for acknowledgement of the state-command, second characters gives information about the state of the external Word-Clock: Y → external Word-Clock is used, N → Word-Clock is disabled.

2.2. X-MADI

Kudos to Lorenzo, the following commands are known for X-MADI:

2.2.1. Read State-Command

Command: *8S#

Answer: *8YY#

2.2.2. MADI-channel routing

Command: *8C CHANNELMODE INPUTCHANNELS OUTPUTCHANNELS#

Example: *8C631B#   for 64-channel-mode with 17-48 for input and 1-32 for output

Answer: *8Y#

"Channel" = 5 for 56 channel MADI-mode, "Channel" = 6 for 64-channel MADI-mode

The possible command-combinations can be:

Input	Output	Channels
0	0	None (?)
1	1	1-32
2	2	9-40
3	3	17-48
4	4	25-56
5	5	33-64

2.2.3. Set fiber-optic / coaxial

Command: *8C601 OPTION#

Example: *8C601O# for fibre-optical

Answer: *8Y#

With "Option" = "C" for coaxial (redundancy) and "Option" = "O" for fibre-optical (priority)

3. General information

3.1.1. Enconding of values as HEX-string

All information regarding card-sizes, position-information, etc. are transmitted as 8-digit HEX-values. So all values have to be converted to HEX-strings.

3.1.2. Used timecode for session-names

The X32 uses a specific timecode based on the year 1980. Day, month, year and time-information are compressed into a 32-bit-value, represented by a 8-digit HEX-string:

For enconding the timecode, the following code can be used:

```
String encode(uint8_t day, uint8_t month, uint16_t year, uint8_t hour,
uint8_t minute, uint8_t second) {
  uint32_t timecode = (uint32_t)((year - 1980)) << 25;
  timecode += (uint32_t)month << 21;
  timecode += (uint32_t)day << 16;
  timecode += (uint32_t)hour << 11;
  timecode += (uint32_t)minute << 5;
  timecode += (uint32_t)second << 1;

  return intToHex(timecode, 8);
}
```

Decoding a given timecode to a nice string can be done as follows:

```
String decode(String timecodeHex) {
  uint32_t timecode = hexToInt(timecodeHex);

  uint16_t year = (timecode >> 25) + 1980;
  uint8_t month =  (timecode & 0x1FFFFFF) >> 21;
  uint8_t day = (timecode & 0x1FFFFF) >> 16;
  uint8_t hour = (timecode & 0xFFFF) >> 11;
  uint8_t minute =  (timecode & 0x7FF) >> 5;
  uint8_t second =  (timecode & 0x1F) << 1;

  return String(day) + "." + String(month) + "." + String(year) + " " +
String(hour) + ":" + String(minute) + ":" + String(second);
}
```