

May 25, 02 16:02

protocol2.txt

Page 1/1

This document is more a dumping ground for the command block format I think the protocol uses.

To see examples of the command block format see protocol.txt and vpl-px21.pdf

all numbers are in hex if i'm referencing a hex number in terms of a decimal number i will give the hex number first and the decimal number second surrounded by ()
the term char that I use comes from the C datatype which can hold one hex code (00-ff)

IMPORTANT! -- usb_control_msg from libusb takes the direction as a part of request type this would normally be 0x00 for out and 0x80 for in

-- Sony thinks otherwise, its 0xc1 for in and 0x41 for out. (directions from the perspective of the device, in means I'm taking input from the NetMD)

//-- standard request

(00/09) - 00 seems to come from the host 09 from the device, start of every binary data chunk.

18 - Peripheral index - in this case the NetMD(18) or 81 which is used for a generic reply code

06/07/43 - Read/write/move - All informational commands seem to use 06 while data changing commands use 07
- actual track movement is done w/ 43

02 20 18- these 3 chars appear in everything except when 20 18 is replaced with 10 10

02 20 10- Track time, codec and bitrate all use this, track info vs track data?

(01/02) - 01 seems to deal with Disc info/Group info while 02 seems to be track info

00 xx - track number(zero based track 1 is 00)

30 00 0a/01 - appear in all disc or track based commands and replies

00 (50/ff) - 50 appears in Set(07 style) commands and ff in get(06 style) commands

00 00 0x - where x is buffer length for set functions

//-- odd man out - 4 char reply from device for buffer size needed to get data

01 is used in the buffer size report

81

<size>

00