Serial Commands

From Dvd-game

The drive accepts 8 debug commmands via the serial interface found on CN302. It reads a single bit from SBI3 when SBT3 goes high.

For every byte(?) a T9 capture B interrupt (0x4C) is generated and handled at [81119,81118,81119] Incoming data goes into a 10 byte ringbuffer at [40EB48,40EB48,40EB48]. [40EB42,40EB42,40EB42] is its write offset.

```
------------------------------------------------------------
SERIAL_CMD_FF:
------------------------------------------------------------
Read memory, does memcopy to 0x40eb54. max 0x20 bytes. Then
transmits over serial if.
Never actually used it, should go over SBO3.

R[0]    FF
R[1]    00
R[2]    scr addr M
R[3]    scr addr L
R[4]    0
R[5]    scr addr H
R[6]    0
R[7]    0
R[8]    nr of bytes to write
R[9]    0


------------------------------------------------------------
SERIAL_CMD_FE:
------------------------------------------------------------
Writes one or two bytes to memory

R[0]    FE
R[1]    00
R[2]    ADDR_M
R[3]    ADDR_L
R[4]    BYTE_1
R[5]    ADDR_H
R[6]    BYTE_2
R[7]    0
R[8]    1-2     nr of bytes to write
R[9]    0


------------------------------------------------------------
SERIAL_CMD_FD:
------------------------------------------------------------
Dumps some interesting memory locations to [40D600,,]

R[0]    FD
R[1]    00
R[2]    z
R[3]    z
R[4]    z
R[5]    z
R[6]    z
R[7]    z
R[8]    z
R[9]    z

 memcpy([40D600,,], [8136,,], 0x48);
 *(u24 *) [40D648,,] = [8136,,];
 *(u8 *) [40D64B,,] = 0;
 *(u24 *) [40D64C,,] = *(u24 *)[817E,,];
 *(u8 *) [40D64F,,] = 0;
 *(u24 *) [40D650,,] = *(u24 *)[8182,,];
 *(u8 *) [40D653,,] = 0;
 *(u16 *) [40D654,,] = *(u16 *)[818E,,];
 *(u16 *) [40D656,,] = *(u16 *)[8120,,];
 memcpy([40D658,,], [819E,,], 0x10);
 memcpy([40D668,,], [81AE,,], 0x10);
 memcpy([40D678,,], [81BE,,], 0x10);
 memcpy([40D688,,], [81CE,,], 0x10);
 *(u16 *) [40D698,,] = *(u16 *)[80F4,,];
 memcpy([40D69A,,], [80B4,,], 0x0c);
 memcpy([40D6A6,,], [80C0,,], 0x18);
 memcpy([40D6BE,,], [80D8,,], 0x04);
 memcpy([40D6C2,,], [8116,,], 0x04);
 memcpy([40D6C6,,], [80FE,,], 0x1e);
 memcpy([40D6E4,,], [40EB7E,,], 0x10);
 *(u16 *) [40D6F4,,] = *(u16 *)[811C,,];


------------------------------------------------------------
SERIAL_CMD_FC:
------------------------------------------------------------
if R4 != 0, drive does not reply anymore (to be more exact it
replies zeroes on dma tranfers) You can bring it back to
life with R4  = 0. Also used on init with R4 = 0. Probably
something like "Enable DI"    *confirm*

R[0]    FC
R[1]    00
R[2]    A5
R[3]    5A
R[4]    0-1
R[5]    FF
R[6]    0
R[7]    0
R[8]    0
R[9]    0


------------------------------------------------------------
SERIAL_CMD_FB:
------------------------------------------------------------
Writes val to [81E5,,] and [40000B,,] ???

R[0]    FB
R[1]    00
R[2]    z
R[3]    z
R[4]    val
R[5]    z
R[6]    z
R[7]    z
R[8]    z
R[9]    z


------------------------------------------------
SERIAL_CMD_F9:
------------------------------------------------
Same as DbgSetReadOptions


------------------------------------------------
```

```
SERIAL_CMD_F8:
-----------------------------------------
Allows some limited HLECommand execution

R[0]    F8
R[1]    00
R[2]    A5
R[3]    5A
R[4]    0-1
R[5]    FF
R[6]    0
R[7]    0
R[8]    0
R[9]    0
```