# StateMachines

### From Dvd-game

the firmware uses some indirect function calls, which look like they were designed as some state machine or so.

Each state machine has a state (called Command_State<n>), and each state machine has different "commands"/"events" (called Command<n>)

They relate to the tables starting at [8DEB6,8DE56,8DF10]

there are 3 tables per state machine. at [8E51C,8E4BC,8E576], there are 7 offsets for the first table (one for each state machine), then 7 offsets for the second table, then 7 offsets for the third table.

Each state machine has a number of states and a number of events. 7 values at [8E57C,8E51C,8E5D6] give the number of states for each state machine (10, 7, 6, 9, 2, 2, 1, 5).

the first table is a lookup table. it has "number-of-states" rows and "number-of-event" columns. so for example for state machine #1, event 1, state 2, you read 0x106. the lower byte is an entry into the third table. the upper byte is always 1. the lower byte will never be 0 or 1 (see third table).

the third table at [8DEB6,8DE56,8DF10] (for State machine #0) are long pointers to the subroutines. the first two entries are always 0 - whyever.

So, if you execute State1(1) with Command_State1 == 2, you take the 6th entry (base 0) in the third table, jumping to [84B10,84B08,84B10], some subroutine which sets ERR_medium_not_present as errorcode (23a00) (because you tried a Read-Command while the cover was open (State 2)).

one example is state machine #1, which is used for DI commands. Depending on some state (Command_State1), a specific DI-Command (decoded into a Command1 in DecodeCommand), executes different code.

There's a subroutine at [83AF0,83AE8,83AF0] which executes a specific event in a specific state machine.

State machines:

## State machine #0: (Mainloop Event state machine)

This state machine's commands are mapped from the mainloop events.

Command events are:

- 0: Cover open event. Sets error codes etc.
- 1: cover closed again. set "Medium may have changed", histate=3
- 2: increments some counter. Or: State2(11).
- 3: ?? very different stuff. Starts CD4 timer! (to 100)
- 4: transfer protocol error
- 5: nop
- 6: State3(1)
- 7: (State3(1)), State7(11)
- 8: some state transition
- 9: ??
- 10: State3(0), .... sometimes even decodes commands. very different stuff.
- 11: State7(12)
- 12:
  - In State 0:
    - Set "PlainState" (80E0) to Command_State1-1 (or 0)
    - clear 1 in "BusyBits" (080DE)
    - Decode and execute Command
    - Handle error code: LastCommandState (i.e. result of command execution)

| retcode | New Command0_State | Desc |
|---|---|---|
| 2 | 4 | ? |
| 3 | 1 | ? |
| 8 | 6 | ? |
| 5 | 2 | ? |
| otherwise | 0 | ClearError, start CD5... (normal case) |

  - in other states: Do nothing (increment counter)
- ...
- 16: CD3 Timer expire. Sets 1/0400 (or something else), somehow depending on the FIFO state
- 17: CD5 timer expire. restarts counter until some condition is true.
- 18: Ident DI command (? whyever)

States are:

- 0: ready for command
- all others: not ready for command.

## State machine #1: (DI Command state machine)

This state machine is used to execute DI commands.

Command states are:

- 0: Ready, sector already read.
- 1: Ready
- 2: Cover Opened
- 3: Disc change detected
- 4: No medium present
- 5: Motor stopped
- 6: disc id not read (initial state)

(the exact difference between 0 and 1 can be discussed somewhere in the audio stuff. It originates from the fact that you can issue the audioconfigure command only as the FIRST command after the read-disc-id.)

Command state can be read using E0 command. 1..6 are mapped to 0..5, 0 is 0.

Commands are:

- 0: CMD12
- 1: CMDE0
- 2: CMDA8
- 3: CMDE2
- 4: CMDE1
- 5: CMDAB
- 6: CMDE3
- 7: CMDE4

- 8: CMDFE
- 9: CMDFF
- 10: error (unknown cmd)
- 11: debug error

## State machine #2..#7:

unknown

Cover Open sets the following states:

Command_State0 = 5

Command_State1 = 2

Command_State2 = 5

Command_State7 = 1

I probably don't have to tell you that these state machines make everything much much MUCH harder to understand. Don't panic.