Pro Action Replay

+Contents

Overview

The Pro Action Replay is a cheat cartridge made for the Master System and Game Gear. It sits between the cartridge and the system and offers the user the ability to search for, and apply, RAM patches.

See also the Game Genie and X Terminator.

User interface

The device has four modes:

- Off: it does nothing, simply passing data to and from the cartridge.
- On: it applies the currently active RAM patches once per frame by intercepting the VBlank handler.
- Code entry: where the user may input cheat codes which tell the device what patches to apply.
- Train: it allows the user to search for a particular variable's RAM location by interrupting the code via the Pause button (SMS) or a button on the device (GG).

Code format

```
00AA AADD
```

Codes are generally given in the form of two 4-digit hexadecimal numbers. The first two digits seem to have no effect; the middle four are the **address** (in the Z80 memory map) of the selected RAM variable; the last two digits are the value which is written there every frame.

Thus, `AAAA` will always be between $C000 and $DFFF and `DD` will be any value from $00 to $FF.

Training mode

The user first plays to a particular point in the game, before activating the "trainer". When starting training mode, the device makes a copy of the system RAM into its own local RAM. The user then restarts the game and plays to another point in the game where the desired parameter is different. Four modes then allow the device to attempt to guess the location containing the parameter in question:

- Known value ("Lives")

  The user enters a decimal value corresponding to the known value shown in-game. The device must deal with issues such as offsets (a value of *n* displayed as *n*+1) and binary coded decimal.

- Known difference ("Timer")

  The user enters the absolute difference between the current value and the previous one. The device must again deal with BCD issues, as well as multi-byte (16-bit or more) variables.

- Unknown value ("Energy bar")

  In this mode, the user selects one of 25%, 50%, 75% and 100% to describe the current variable value. The device must deal with different size variables again as well as approximate matches.

- State change ("Start/Change")

  This deals with variables that are defined by changes of state. The user tells the device whether the current state is the same as, or different to, the starting point.

In each case the software must use the information given to attempt to iteratively narrow down the list of potential matches until it (ideally) reaches one. More precise methods (such as the first) allow this to be done faster. The user can then make a note of the successful codes.

How it works

The device contains hardware to map its own internal ROM (and RAM) in place of the game, which can then execute code to access system RAM and perform additional tasks when the system expects to be accessing the game. By watching the address lines, or the INT and NMI cartridge lines, it can detect interrupts and intercept these, by mapping itself in, performing its tasks, and then mapping the game ROM back into memory and passing control to its handlers for those routines.

By intercepting the VBlank interrupt, it can write the chosen values to RAM once per frame, thereby offering cheating capabilities. By intercepting the NMI, it can latch onto the Pause button handler for activating its code for "training". (On the GG, it instead maps itself in on reset, and provides a hardware button to trigger a reset.)

It is unclear whether the device is able to differentiate between VBlank interrupts and HBlank interrupts. One might expect it to cause problems with timing-sensitive games.

The hardware has two "registers" that affect its mapping of its ROM to the Z80 address space. Writes to $2000 cause the upper 16KB of its ROM to be mapped when activated. The value written does not matter. This upper 16KB contains the code to apply RAM patches when an interrupt happens. Writes to $6000 cause the PAR ROM to be deactivated; again the value written does not matter. Thus, it has several modes of operation:

1. When disabled, the PAR ROM is never active.
2. When enabled, and the game boots, the lower 16KB PAR ROM is activated automatically.
3. When booting a game from the menu, the PAR ROM is deactivated but the upper 16KB is selected.
4. When an interrupt happens, and the switch is in the "active" position, the (upper) PAR ROM is activated, in order to apply any cheat codes.
5. When returning to the PAR menus in "training mode" (via a pause interrupt hook or a reset hook on GG), the lower 16KB PAR ROM is activated.

Additional information

- GG PAR manual
- GG PAR codebook

The device interferes with the Gear to Gear Link mode in some (or all) games.

The PAR only works intermittently (on about 2/3 of games) when used with a Game Gear with a BIOS.

Codemasters games will not work with a PAR. It may not be connecting the additional cartridge lines needed by the Codemasters mapper hardware.

Easter Egg

By holding Button 2 and activating the code menu the PAR will display the credits and the compilation date and version number of the ROM. On SMS it will also add the code 00C82606 to the parameters menu.

Game Gear credits

```
Program -W.H.Beckett
Design  -W.H.Beckett
Hardware-C.R.Harding
Rev. Eng-W.H.Beckett
   &    -C.R.Harding
C.O.    -M.J.Connors
Ind. Esp-M.  Wallace
2 I.C.  -I.  Ryles
Testing -Bazz &
        -Trev

   18/05/93 v1.01
  Datel Electronics
```

Master System credits

```
     Program   ... W.H.Beckett
     Design    ... W.H.Beckett
     Hardware  ... C.R.Harding
     Rev. Eng. ... W.H.Beckett
        &      ... C.R.Harding
     C.O.      ... M.J.Connors
     Ind. Esp. ... M.  Wallace
     2 I.C.    ... I.  Ryles
     Testing   ... Bazz &
               ... Trev
```

```
    Software 04/3/93 v1.02
(C) Datel Electronics Ltd.
```

- Back to Memory System index
- Back to Development index

Return to top

0.072s