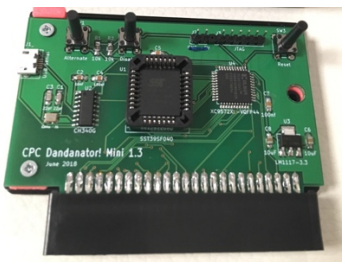
- CPC Dandanator mini – Development & Emulation documentation -

Hello and thank you for your interest in the CPC Dandanator mini cartridge. This document describes its internal functionality in detail, as well as implementation tips for emulating the hardware. Remember that this is a Public Domain project and all code/hardware/description is free to use with no restriction whatsoever.

Part 1: Hardware description

The CPC Dandanator mini is a hardware cartridge for Amstrad CPC computers and compatible variants such as Schneider/Aleste computers (note these computers would need a physical adaptor for plugging-in the cart). Externally it has a female edge 25x2 connector, three buttons and a micro-usb port.
Internally, it's built around a Xilinx xc9572xl CPLD (44 VQFP) and an SST39SF040 (PLCC32) 512kx8 flash EEPROM memory. USB protocol is taken care by a dedicated CH340G chip.
The board is designed to fit into a Supertronic PP6N case, by using three small 3D-printed parts to hold it firmly.
All three buttons are Alps SKHHDHA010 or TE-Connectivity FSM16JH, SPST vertical switches, soldered to the PCB and with a 17mm high stem to reach out to the top of the enclosure.
While the right button is a direct CPC Reset button, left and middle buttons perform actions during boot or reset.



Part 2: Mode of operation

MEMORY PAGING

The cartridge has a 512KB flash EEPROM memory that is divided into 32 slots of 16KB each. Primary functionality is to page between these slots in different memory positions.

At any point in time, none, one or two slots may be mapped to any memory segment of the CPC. A segment is an Amstrad CPC contiguous memory area chunk of 16KB that is at 0x0000, 0x4000, 0x8000 or 0xC000.
When a slot is mapped into a segment, reads from that memory segment are taken from the cartridge, regardless of the CPC ROM/RAM configuration. Writes to a memory segment allocated by a slot are always passed down to the internal RAM.
Each mapped memory slot is called a "zone". So CPC Dandanator has two zones. Zone 0 may be mapped to 0x0000 or to 0x8000 and zone 1 to 0x4000 or 0xC000.

USB OPERATIONS

USB protocol is managed by a dedicated CH340G chip, implementing the CDC profile (Serial port). All communication between this chip and the CPC relies on bitbanging.

Reading from USB:
By sending a command to the CPC Dandanator, the serial input is enabled and the LD A,(HL) stops returning (HL) and returns the status of the serial port in bit 0. Either 1 for idle line or 0 for active. All other bits remain at bus-idle, high impedance. A code library for reading at 57.600,N,8,2 is provided in the code section of the document.
It's important to disable serial input when not in use. Otherwise, "LD A,(HL)" instruction will not work as expected.
Note that LD A,(IX+n) - LD A,(IY+n) - BIT 7,(HL) - BIT 7,(IX+n), BIT 7, (IY+n) are also affected by this bus hacking. Do not use these opcodes while in USB Read mode.

Writing to USB:
Writing to USB is always possible and doesn't require enabling. This is also accomplished by bitbanging. A command changes the state of the output line between 1, idle and 0, active. A code library for sending at 57.600,N,8,2 is provided in the code section of the document. Note that, because of the current code, writing to USB also disables reads, although this is not mandatory and may be changed at will.

FLASH OPERATIONS

The contents of the flash memory can be overwritten by software from the CPC. To do so, a command must be sent to enable writes to the chip. Afterwards, a JEDEC sequence for sector erasing and sector writing must be submitted to the memory chip. Please check the part datasheet http://ww1.microchip.com/downloads/en/DeviceDoc/20005022C.pdf for detailed information.
A code library for erasing a sector and writing a sector is provided in the code section of this document.

DELAYED ACTION

While the CPC Dandanator executes the commands right after the fetch-cycle of the command instruction, some configurations can be delayed until a 0xC9 opcode (RET) is fetched (note that multibyte instructions with a 0xC9 will also trigger this. Some examples are "SET 1,C" (0xCBC9) or some IX, IY bit operations such as "SET 1, IX+n, C").
The main reason for this feature is the atomic action of transfer control of the PC to an external program and disable/enable/configure the cartridge.
The execution of these actions occurs right after the Z80 0xC9 operand fetch.
Delayed configuration parameters are:

- Disable further commands until reset.
- Trigger FollowRomEnable, so Dandanator will only be enabled if a Rom is selected in the CPC. Useful for low-high rom substitution (poor-man rombox).
- A15 values for Zone 0 and Zone 1 so you can change them between segments
- Enable status for Zone 0 and Zone 1 -> Zones may remain enabled or get disabled.

Part 3: CPLD Commands

The CPLD receives commands by sniffing M1 cycles (operation fetch) of the Z80.
A command is then recognized by a trigger prefix. The selected prefix is 0xFDFDFD as seen from the CPLD, (actually the prefix is any number, greater than 2, of 0xFD, but the above prefix is strongly recommended).
Right after the prefix, a 2 byte FDxx instruction is the actual command. This is why, from an ASM stand point, the actual prefix is 0xFDFD and then then 0xFDxx instruction completes the three-FD sequence.
This prefix has a couple of interesting effects:

a) It's an actual Z80 prefix itself, so no operation is executed by the Z80 and no RAM or registers are changed. Only PC and R are affected as defined.
b) Interrupts and NMIs are not served until the end of the full Dandanator command, so there is no need to DI in advance.

PAGING COMMANDS

Zone 0 and Zone 1 use different commands for paging:

- Zone 0 Command: Trigger + **LD (IY+0),B**
- Zone 1 Command: Trigger + **LD (IY+0),C**

Registers B and C are read by the CPLD as follows:

- Bits 4-0: Slot to assign to zone
- Bit 5: Eeprom Chip Enable for zone. '0' is enabled, '1' is disabled.
- Other bits: ignored.

These commands overwrite a RAM byte. Keep this in mind and point IY to a suitable scratch address. Also IY+0 is recommended but not mandatory, the value n in (IY+n) is ignored by the CPLD.

The following code selects slot 15 for zone 0 and enables it. It also selects slot 17 for zone 1 but disables the mapping of this zone.

```
LD B, $0F           ; 0bxx001111 : underlined bit is eeprom chip enable
LD IY, SCRATCHBYTE  ; This byte in ram will be overwritten
DEFB $FD, $FD       ; Prefix for the Z80
LD (IY+0), B        ; Zone 0 command -> 0xFD
LD C, $31           ; 0bxx110001 : underlined bit is eeprom chip enable
DEFB $FD, $FD       ; Prefix for the Z80
LD (IY+0),C         ; Zone 1 command
```

CONFIGURATION COMMANDS

To configure the behaviour of the CPC Dandanator, the CPLD reads the "configuration command", **LD (IY+0),A**.

This table summarizes the format of the command (A value):

when bit 7=0:	b4-b3:	FollowRomEnable zone 0 slot lower bits: slots 28,29,30 and 31
	bit 2:	Out bit to USB (Serial Bitbanging): "1" Idle.
	bit 1:	EEprom write enable "1" or disable "0"
	bit 0:	Serial port ena/dis - When enabled, LD A,(HL) returns serial RX in bit 0.
when bit 7=1:	bit 6:	Wait for "RET" (0xC9) to execute actions
	bit 5:	Disable Further Dandanator commands until reset
	bit 4:	Enable FollowRomEn on RET (only read if bit 6 = 1)
	b3-b2:	A15 values for zone 1 and zone 0. Zone 0 can be at 0x0000 or 0x8000, zone 1 can be at 0x4000 or 0xC000
	b1-b0:	Status of EEPROM_CE for zone 1 and zone 0. "0": Enabled, "1" Disabled.

OTHER SPECIAL COMMANDS

RET (0xC9) and **LD A,(HL)** (0x7E) and all other multi-byte commands including an M1 cycle with these values are recognized as commands in special situations:

- RET – When a "Wait for RET" has been previously sent and no other command has been issued since.
- LD A, (HL) – When USB Read is enabled.

No Trigger is needed for these special commands.

Part 4: Boot Sequence

These values apply to boot and/or reset sequence.

Normal boot:

- Zone 0 enabled on slot0 and segment 0x0000.
- Zone 1 disabled on slot0 and segment 0x4000.
- USB RX off.
- USB TX on idle state : '1'.
- Eeprom writes off.
- FollowRomEnable off.

Boot with left button pressed. Same as normal boot but:

- Zone 0 enabled on slot31 and segment 0x0000. → boot on slot31, useful for hardware test roms.

Boot with middle button pressed. Same as normal boot but:

- Zone 0 disabled on slot0 and segment 0x0000. → boot internal CPC firmware/basic.
- This button has prevalence over left button, so if both are pressed during reset/boot, the CPC will boot to internal firmware/basic.

Part 5: Code Libraries

Code libraries are supplied alongside this document as Z80 ASM files. All files are compiled using SJASM 1.07, but adaptation to PASMO or any other compiler should be trivial.
In addition, the VHDL definition used for the CPLD is also provided.

- **ddntr_macros.asm** : macros for commands
- **usbserial56k_CPC_v2.asm** : functions for USB read and write.
- **sstwriter_CPC_v3.asm** : functions for erasing and writing SST Flash sectors.
- **CPCDandanator.vhd** : VHDL definition of the CPLD.
- **Menusystem.rom** : A game menu with several CPC games for running in a Dandanator.

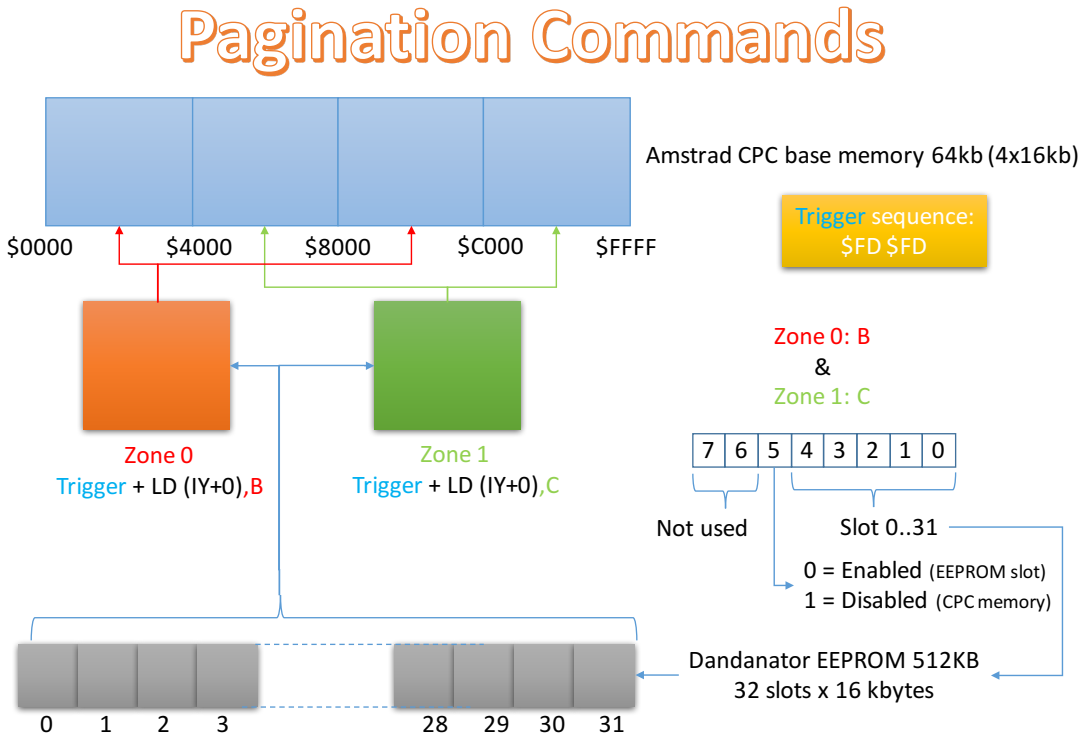
Part 6: Acknowledgements

Many, many thanks to:

- **Manuel Teira "overCLK"** for the Java tools and fine tuning of USB communication.
- **Mario Jiménez "mad3001"** for his Z80 code contributions and the quick reference of commands graphics on this document.
- **Javier Chocano "Habi"** for adapting his personal CPC Emulator to support the cartridge and for his help on VHDL.
- **Francisco Javier Velasco "Pagantipaco"** for the mode0 logo.
- **Javier García Navarro (4Mhz)** for selecting the CPC Dandanator as their game distribution media.
- **Miguel Angel Rodríguez Jodar "mcleod_ideafix"** for providing his CPC hardware test rom.
- **Einar Saukas** for his ZX7 compressor.
- **Juan Carlos González Amestoy** for supporting the CPC Dandanator in his RetroVirtualMachine emulator and for his fine tuning tips.
- **César Hernández Baño** for adding support of the cartridge to his ZesarUX emulator.
- **Kevin Thacker** for emulating the cartridge on his Arnold emulator.
- **CPC Community**: cpcmaniaco, alt, benkenobi, merlinkv, kikems et al for their support and their welcoming of the device.

Part 7: Quick Reference of Commands

Pagination Commands



Configuration Commands