



MNT Research GmbH
 Fehlerstr. 8, 12161 Berlin
 WEEE: DE 33315564
<https://mntre.com>

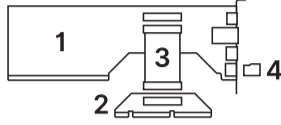
MNT ZZ9000 R-3 Quickstart Manual

Dear MNT Customer,

Thank you for purchasing the MNT ZZ9000 Graphics Coprocessor Card (with Ethernet and USB Storage) for Commodore Amiga computers. This product was made possible by your support.

Package Contents

- MNT ZZ9000 Zorro card (1) with mounting bracket and 128MB MicroSD card (4) containing ZZ9000OS
- MNT ZZ9000CX ECS/AGA Video Slot adapter (2)
- Video Slot ribbon cable for connecting both cards (3)
- This manual



Warranty

MNT Research covers the warranty to the extent required by law for the unmodified product. ZZ9000 is based on a reconfigurable Xilinx ZYNQ Z-7020 FPGA that can be updated by the user and reprogrammed to enable new functionality. Warranty does not cover user supplied firmware or any modifications made to the card, nor does it cover any damage by inserting it into a computer or other user supplied device. Use at your own risk!

Declaration of Conformity

MNT ZZ9000 may only be operated in a shielding steel enclosure. MNT Research assures that the included product complies with the requirements of the following applicable European Directives and DIN standards:



EMC Directive 2014/30/EU, Low Voltage Directive 2014/35/EU
 RoHS Directive 2011/65/EU
 DIN EN 55022:2011-12
 DIN EN 55024:2016-05
 DIN EN 61000-6-1:2007
 DIN EN 61000-6-3:2007/A1:2011/AC:2012
 DIN EN 61000-3-2:2014
 DIN EN 61000-3-3:2013

Get updates and source code at:
<https://mntre.com/zz9000>

Need help? Write to
zz9000@mntre.com

Join our IRC channel **#mnt-amiga** on
irc.freenode.net

Lukas F. Hartmann, Managing Director
 MNT Research GmbH
 November 17th, 2020

Running ARM Software

ZZ9000 has two 32-bit ARM Cortex-A9 processor cores that run at 666 MHz. The system has access to 1 GB of DDR3 memory in total. A part of this memory is directly shared with the Amiga's chipset and 68000 family processor via the Zorro Bus. The Amiga RTG screens also occupy the same memory space, so both ARM and Amiga can draw pixels into graphics memory or read from it.

Normally, a mini operating system called ZZ9000OS is running on ARM Core 1, and ARM Core 2 is idle. ZZ9000OS has the following functions:

- Polls the Zorro bus in the FPGA fabric for events (writes, reads, or Amiga reset)
- Receives and transmits Ethernet packages
- Controls the DVI transmitter
- Controls the USB port
- Executes accelerated RTG drawing functions like fills, scrolling, text, patterns

You can run custom code on ARM Core 2 by loading it into a place in shared memory and asking ZZ9000OS to jump into the code. On the ZZ9000 website (<https://mntre.com/zz9000>) you can download example programs and a simple SDK for writing your own ARMv7 applications, as well as the sources for ZZ9000OS.

Updating FPGA Firmware and ZZ9000OS

Development of ZZ9000 has taken almost two years, and it was our priority to get the hardware into your hands as soon as possible while delivering a good feature set to begin with. But ZZ9000 is highly extensible and open source. In addition to us delivering new features for the card, we believe that the hackers and makers among you will come up with fantastic new software and ideas to run on ZZ9000.

The ZYNQ on ZZ9000 boots from the inserted MicroSD card. This card needs to be MBR partitioned and FAT32 formatted. The system looks for a file called BOOT.bin which contains the ZYNQ FSBL (first stage boot loader) that in turn loads the FPGA image and ZZ9000OS (in ELF format). To update the card, you simply have to replace the BOOT.bin file with a new one.

ARM Address	Amiga Address*	Area
\$0010_0000	N/A	DDR3 Start, ZZ9000OS Code
N/A	\$0000_0000	ZZ9000 ARM Registers
N/A	\$0000_1000	ZZ9000 FPGA Registers
N/A	\$0000_2000	Ethernet and USB Buffers / Scratch Space
\$0020_0000	\$0001_0000	Framebuffer Memory Start
\$005F_0000	\$0040_0000	Zorro 2 Accessible Space End (4 MB)
\$0300_0000	\$02E1_0000**	ARM Application and General Purpose Shared Space Start
\$101F_0000	\$1000_0000	Zorro 3 Accessible Space End (256 MB)
\$4010_0000	N/A	DDR3 End (1 GB)

*Relative to card address

**Currently unreachable on Zorro 2 machines, update coming

Hardware Installation

1. Connect the Video Slot ribbon cable by inserting one end into the 34 pin socket of ZZ9000 and the other end into the 34 pin socket of ZZ9000CX.

2. Insert ZZ9000 into any of the Zorro slots of your Amiga 2000/3000/4000.

3. Insert ZZ9000CX into the Video Slot of your Amiga. In Amiga 2000, the Video Slot is on the right side of the power supply. In Amiga 3000, the Video Slot is located next to the top Zorro slot on the daughterboard. In Amiga 4000, the Video Slot is next to the bottom Zorro slot on the daughterboard.

There are two distinct firmware (BOOT.bin) files for Zorro II and Zorro III support. ZZ9000 ships with the firmware that you selected when ordering. To change it, just download a new firmware file and copy it to the MicroSD card inserted in ZZ9000.

Software Installation

1. Install the RTG (ReTargetable Graphics) system. There are two options:

- Picasso96 on Aminet: <http://aminet.net/package/driver/video/Picasso96>
- P96 2.x by Individual Computers (recommended). To purchase a license, first register an account in their shop. More information: <http://wiki.icomp.de/wiki/P96>

When installing Picasso96/P96, select only the **Altai**s card. The ZZ9000 Installer will copy it to create its own Devs:Monitors/ZZ9000 driver file.

2. ZZ9000 implements a SANA-II compatible network card that you can use to go online with TCP/IP stacks such as Miami or Roadshow. **Make sure you have such a TCP/IP stack** installed before installing the network driver.

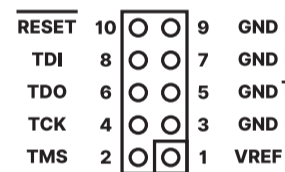
3. Double click "Install ZZ9000" from the ZZ9000 Install package available from our website or on floppy disk from our online shop. The ZZ9000 Installer will guide you through all steps necessary to set up the graphics and network functions.

4. After rebooting, you will be able to select high resolution 8/16/32 bit screen modes using ZZ9000's display engine. Native Amiga chipset screen modes will be automatically scan-doubled and flicker-fixed via ZZ9000CX.

5. You can use a USB stick to exchange files between your Amiga and other computers via ZZ9000USBStorage.device included in the Installer. On your PC, create partitions (<4GB recommended) and format the stick on your PC to FAT32. Plug the stick into ZZ9000's USB port before switching on the Amiga (or resetting). Use **fat95** and **MountDos** to easily mount the USB partitions:

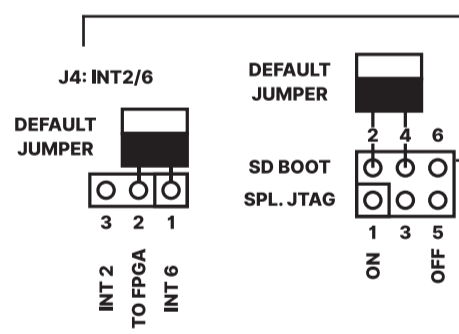
- <http://aminet.net/package/disk/misc/fat95> (Copy fat95 to L:)
- <http://aminet.net/package/disk/misc/MountDos12> (Copy mountdos to C:)

Enter in a Shell: `mountdos ZZ9000USBStorage.device fat95 mount`

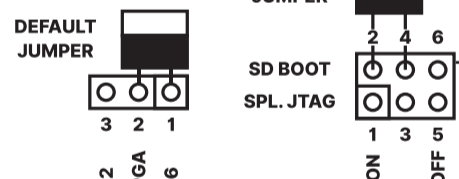


Jumpers and Headers

J1: JTAG. You can debug and reprogram both FPGA and ARM through JTAG via a Xilinx Platform Cable or similar and using free Xilinx Vivado Webpack software.



J4: Interrupt select. Connects either Amiga interrupt INT2 or INT6 to the FPGA. Leave the jumper on INT6 to use Ethernet.



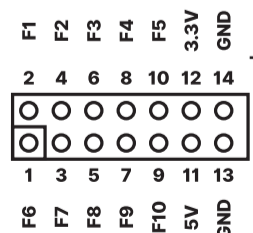
J13: Boot select. 2+4 = Boot from MicroSD card. This is the default and should normally not be changed. If you remove this jumper, the system will boot from a small QSPI flash ROM on the ZYNQ module. A minimal Linux distribution is installed for testing. Don't do this with ZZ9000 inside of your Amiga, because the I/O pins of the FPGA won't be configured without a correct boot image.



J12: UART. Connect a USB-to-UART cable to RX, TX and GND pins (115200 baud) to see ZZ9000's debug output.



J6: Audio. Breaks out the the raw and filtered audio signals of the video slot for use in future sound expansions.



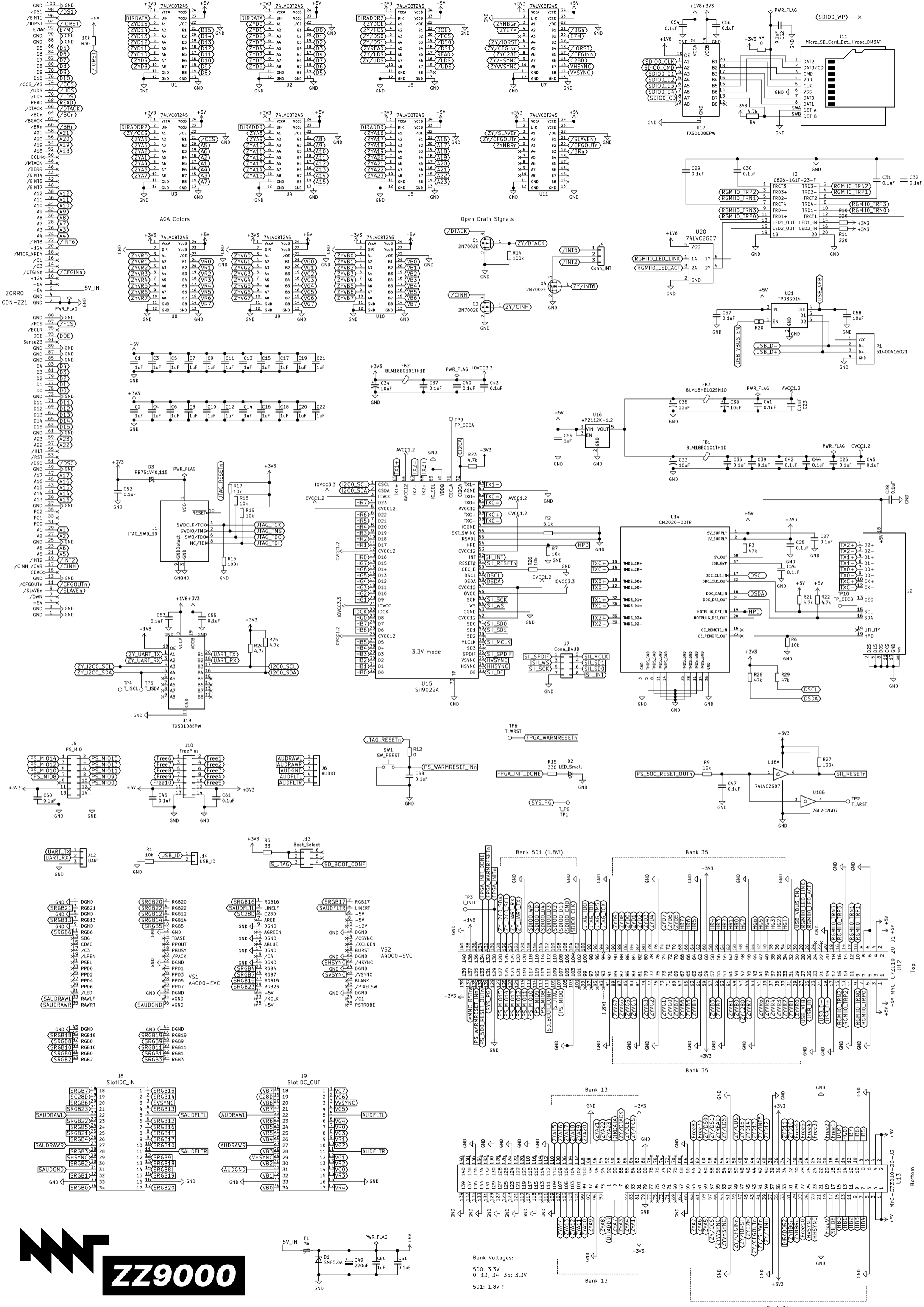
J10: FPGA Expansion. 10 additional unused FPGA inputs/outputs for later use.

J5: MIO. Breaks out 8 unused ZYNQ MIO pins. Reserved for later use, not shown here.

J7: Digital Audio Input for the video encoder chip. Reserved for later use, not shown here. Introduced in R-2.

Your unique Ethernet MAC address is: **68:82:F2:01:02: FF**

MAC addresses for ZZ9000 are sponsored by **grandcentrix**. Visit <https://grandcentrix.team> for exciting career opportunities.



- GND 100 → GND
- /D51 98 → /D51
- /EINT1 96 → /IORST
- /IORST 94 → /IORST
- E7M 92 → E7M
- GND 90 → GND
- GND 88 → GND
- D5 84 → D5
- D6 82 → D6
- D7 80 → D7
- D8 78 → D8
- D9 76 → D9
- /CCS_AS 74 → /CCS
- /UDS 70 → /UDS
- /LDS 70 → /LDS
- READ 68 → READ
- /DTACK 66 → /DTACK
- /BGn 64 → /BGn
- /BGACK 62 → /BGACK
- /BRn 58 → /BRn
- A20 56 → A20
- A19 54 → A19
- A18 52 → A18
- ECLK 50 → ECLK
- /MTACK 48 → /MTACK
- /BERR 46 → /BERR
- /EINT4 44 → /EINT4
- /EINT5 42 → /EINT5
- /EINT7 40 → /EINT7
- A12 38 → A12
- A11 36 → A11
- A10 34 → A10
- A9 32 → A9
- A8 30 → A8
- A7 28 → A7
- A3 26 → A3
- A4 24 → A4
- /INT6 22 → /INT6
- /INT6 20 → /INT6
- /MTRC_XRDY 18 → /MTRC_XRDY
- /C3 16 → /C3
- /C13 14 → /C13
- /CFGInn 12 → /CFGInn
- +12V 8 → +12V
- +5V 6 → +5V
- ZORRO CON-Z21 GND 2 → GND
- GND 99 → GND
- /FCS 97 → /FCS
- /BCLR 92 → /BCLR
- DOE 91 → DOE
- Sense23 GND 89 → GND
- GND 87 → GND
- GND 85 → GND
- D4 84 → D4
- D3 82 → D3
- D2 80 → D2
- D1 78 → D1
- D0 76 → D0
- GND 74 → GND
- D11 69 → D11
- D12 67 → D12
- D13 65 → D13
- D14 63 → D14
- D15 61 → D15
- GND 61 → GND
- A23 59 → A23
- A22 57 → A22
- /HLT 55 → /HLT
- /RST 53 → /RST
- /D50 51 → /D50
- GND 49 → GND
- A17 47 → A17
- A16 45 → A16
- A15 43 → A15
- A14 41 → A14
- A13 39 → A13
- GND 37 → GND
- FC2 35 → FC2
- FC1 33 → FC1
- FC0 31 → FC0
- A1 29 → A1
- A5 19 → A5
- /INT2 17 → /INT2
- /CINH_OVR 17 → /CINH
- CDAC 15 → CDAC
- GND 13 → GND
- /CFGOUTn 11 → /CFGOUTn
- /SLAVEn 9 → /SLAVEn
- /OWN 5 → /OWN
- GND 3 → GND
- GND 1 → GND

- GND 99 → GND
- /FCS 97 → /FCS
- /BCLR 92 → /BCLR
- DOE 91 → DOE
- Sense23 GND 89 → GND
- GND 87 → GND
- GND 85 → GND
- D4 84 → D4
- D3 82 → D3
- D2 80 → D2
- D1 78 → D1
- D0 76 → D0
- GND 74 → GND
- D11 69 → D11
- D12 67 → D12
- D13 65 → D13
- D14 63 → D14
- D15 61 → D15
- GND 61 → GND
- A23 59 → A23
- A22 57 → A22
- /HLT 55 → /HLT
- /RST 53 → /RST
- /D50 51 → /D50
- GND 49 → GND
- A17 47 → A17
- A16 45 → A16
- A15 43 → A15
- A14 41 → A14
- A13 39 → A13
- GND 37 → GND
- FC2 35 → FC2
- FC1 33 → FC1
- FC0 31 → FC0
- A1 29 → A1
- A5 19 → A5
- /INT2 17 → /INT2
- /CINH_OVR 17 → /CINH
- CDAC 15 → CDAC
- GND 13 → GND
- /CFGOUTn 11 → /CFGOUTn
- /SLAVEn 9 → /SLAVEn
- /OWN 5 → /OWN
- GND 3 → GND
- GND 1 → GND

- GND 4 → GND
- PS_MIO14 1 → PS_MIO14
- PS_MIO15 2 → PS_MIO15
- PS_MIO12 3 → PS_MIO12
- PS_MIO10 4 → PS_MIO10
- PS_MIO11 5 → PS_MIO11
- PS_MIO8 6 → PS_MIO8
- PS_MIO9 7 → PS_MIO9
- PS_MIO0 8 → PS_MIO0
- PS_MIO1 9 → PS_MIO1
- PS_MIO3 10 → PS_MIO3
- PS_MIO4 11 → PS_MIO4
- PS_MIO5 12 → PS_MIO5
- PS_MIO6 13 → PS_MIO6
- PS_MIO7 14 → PS_MIO7
- GND 15 → GND
- GND 16 → GND
- GND 17 → GND
- GND 18 → GND
- GND 19 → GND
- GND 20 → GND
- GND 21 → GND
- GND 22 → GND
- GND 23 → GND
- GND 24 → GND
- GND 25 → GND
- GND 26 → GND
- GND 27 → GND
- GND 28 → GND
- GND 29 → GND
- GND 30 → GND
- GND 31 → GND
- GND 32 → GND
- GND 33 → GND
- GND 34 → GND

- GND 4 → GND
- SRGB18 5 → SRGB18
- SRGB19 6 → SRGB19
- SRGB7 7 → SRGB7
- SRGB8 8 → SRGB8
- SRGB10 9 → SRGB10
- SRGB11 10 → SRGB11
- SRGB1 11 → SRGB1
- SRGB2 12 → SRGB2
- SRGB3 13 → SRGB3
- SRGB4 14 → SRGB4
- SRGB5 15 → SRGB5
- SRGB6 16 → SRGB6
- SRGB9 17 → SRGB9
- GND 18 → GND
- GND 19 → GND
- GND 20 → GND
- GND 21 → GND
- GND 22 → GND
- GND 23 → GND
- GND 24 → GND
- GND 25 → GND
- GND 26 → GND
- GND 27 → GND
- GND 28 → GND
- GND 29 → GND
- GND 30 → GND
- GND 31 → GND
- GND 32 → GND
- GND 33 → GND
- GND 34 → GND

