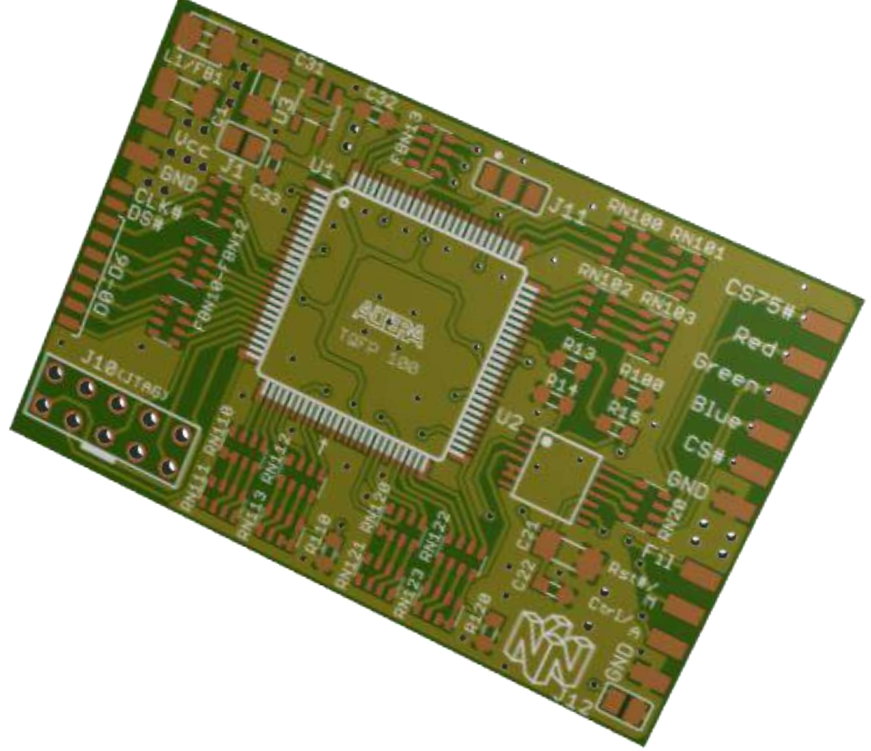
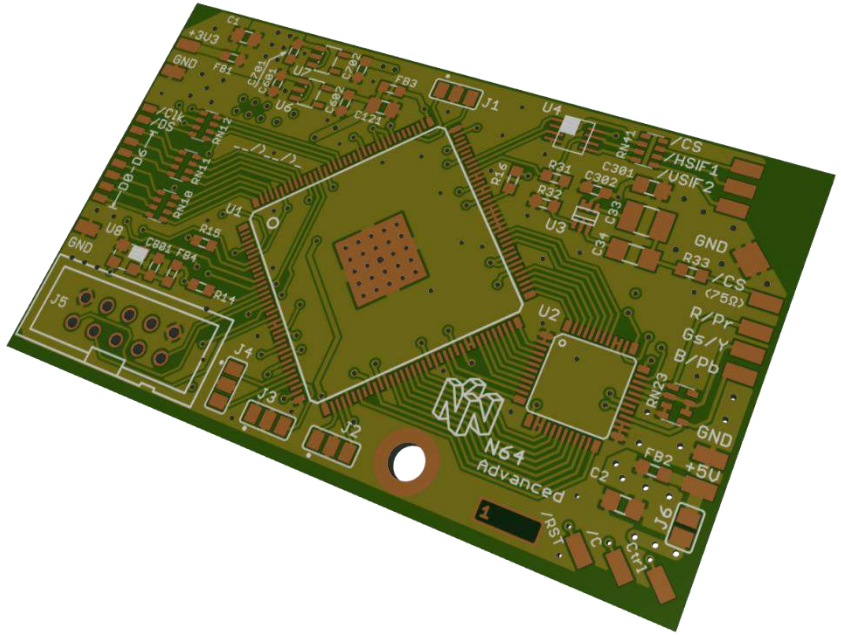
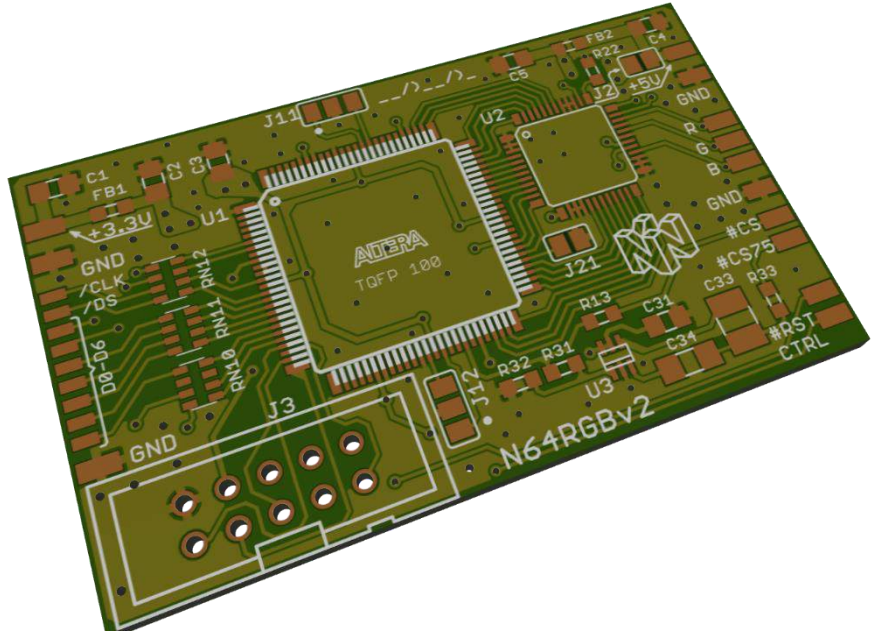
N64 Advanced
and
N64RGB Version 1 and 2



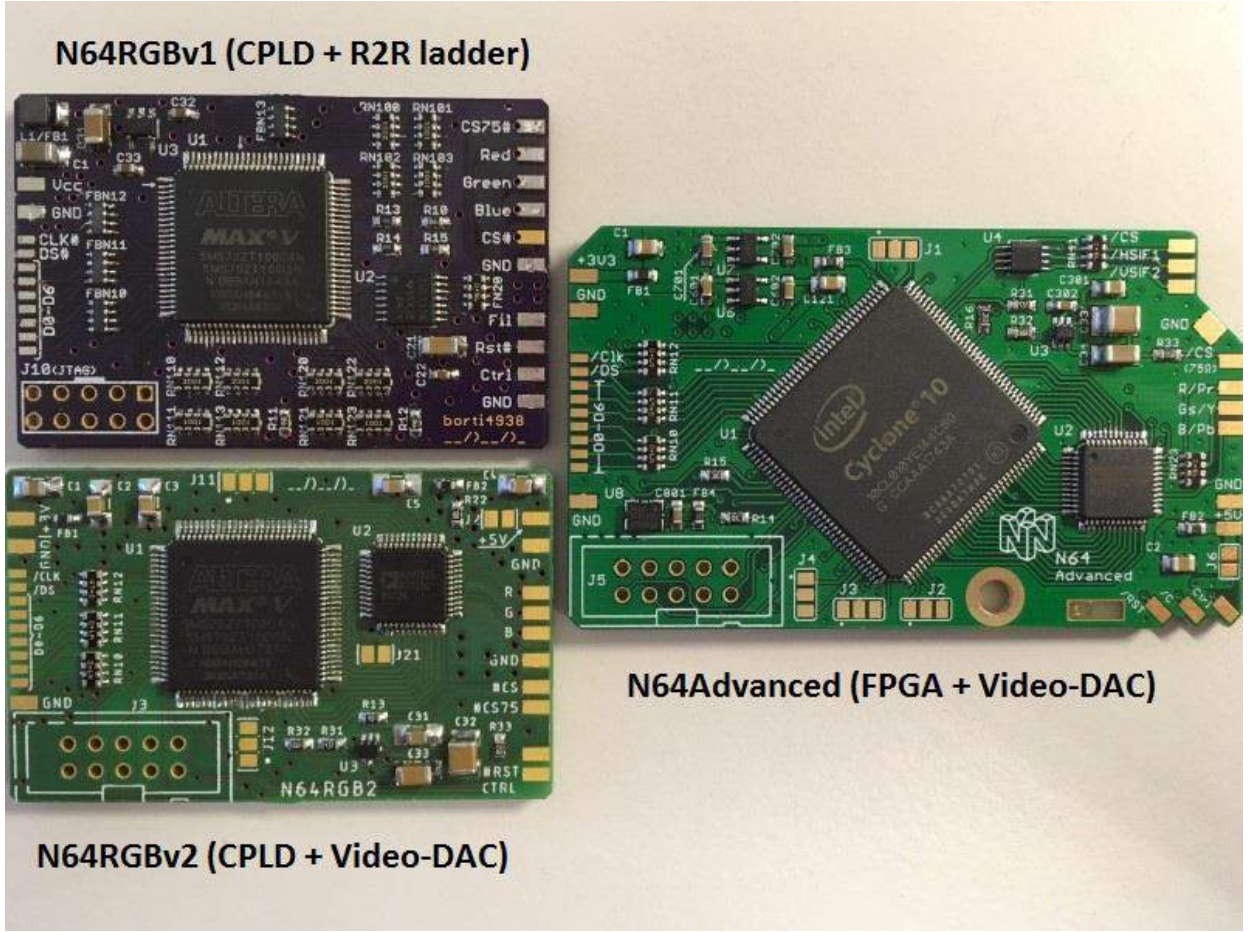


Installation and
Setup Guide



Content

- Open the Console
- Preparation
- Solder Work
  - Console
  - Modding Kit
- Settings (Solder Jumper)
- Cable Setup
- Controller – In Game Routines
- Firmwareupdate

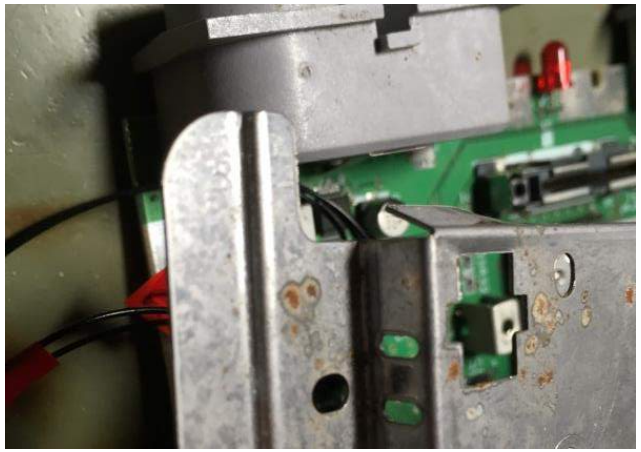


Open the Console

- Remove Jumper Pak / Expansion Pak
- Remove screws from bottom side of the console (needs 4.5mm game bit tool)
- Lift up the top housing
- Remove marked screws from inside (in some later consoles heat sink is designed differently)
- Pull out the mainboard and remove heat sink and RF shield
- **Hint:** Now you have a good chance to clean up your N64 shell under hot water.

Picture by Zerberus (circuit-board.de user)

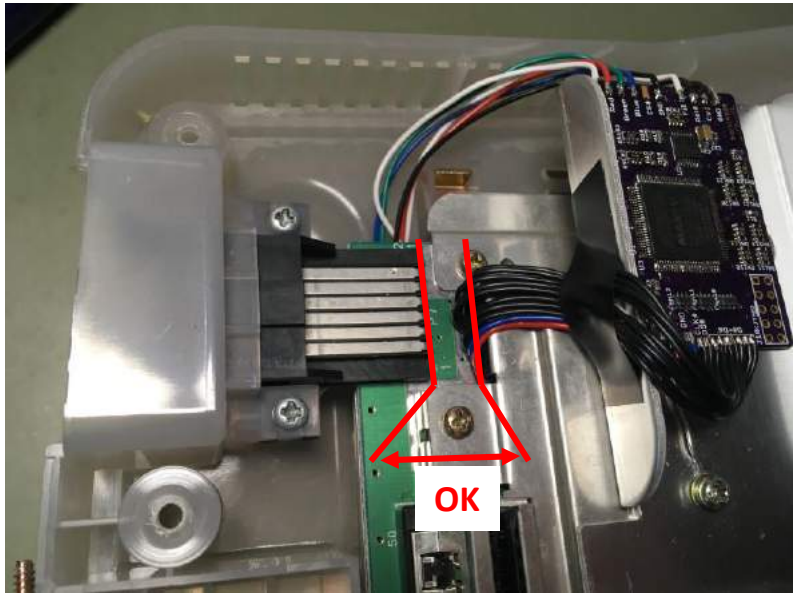
Preparation – Top RF shield

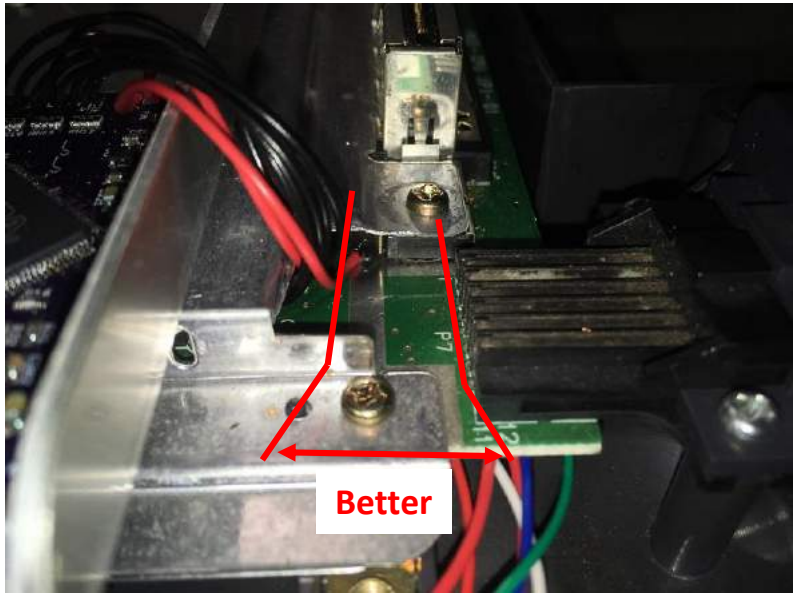


- Next to the MultiOut and the cartridge slot
  - Locate the tap in the RF shield closing the gap to the N64 mainboard
  - Bend away or remove this tap or even cut off a small piece out of the RF shield as shown here
- At the front side between reset button and controller port:
  - Slightly bend off the tap here
- This helps later to fit the connection wire

Preparation – Top RF shield

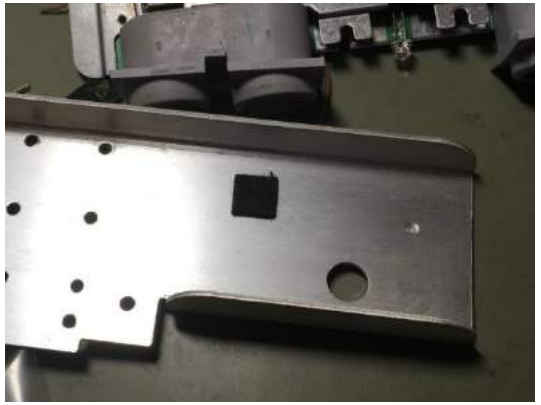
-- Interference issue--



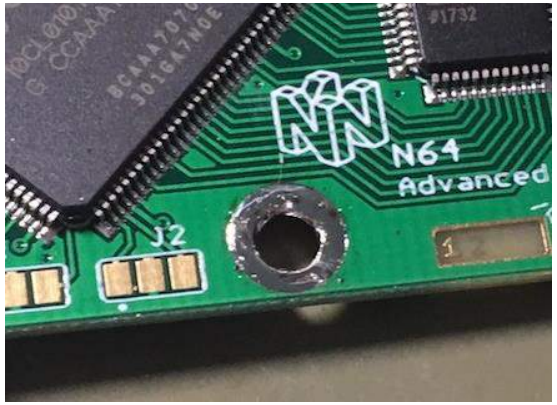


- Later, the data lines as well as the clock will be routed close to the MultiAV
- The distance between both is a crucial issue as your wires act as interferer
- Hence, have tap bend might be ok, but it is better to cut some piece of the RF shield away to further increase the distance between MultiAV and data lines

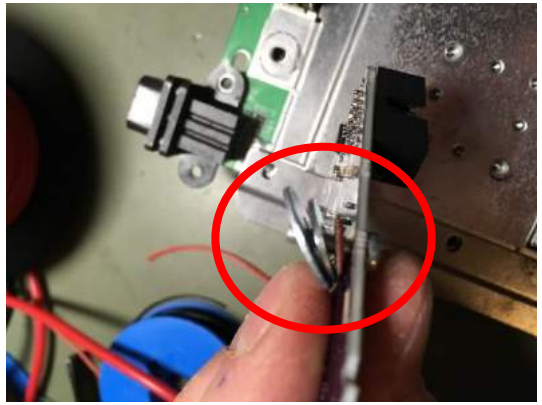
Preparation – Mount the PCB (N64A only)



- Put a small piece of a flat isolationg foam or glider under the top left of the PCB (OPTIONAL)



- Apply some amount of solder to the mounting hole
- This helps later to have a better connection to the screw



Mounting material:

- Screw
  (M3 diameter,
  min. 10mm, max. 14mm)
- 3x washer below PCB:
  - isolating or small outer diameter
    (three traces are close to the mounting hole)
  - a slightly larger washer
  - a large washer
    (such that this washer does not fit through the mounting hole of the heat sink)

Preparation – Mount the PCB (N64A only)

- Place the PCB on top of the heat sink

- Fasten screw from the bottom side of the heat sink
- Use another washer and a (locking) nut
- **Attention:**
  - Do not place the washer into the RF shield area!
  - Do not stress the PCB to much!

Preparation – Mount the PCB (N64RGB only)

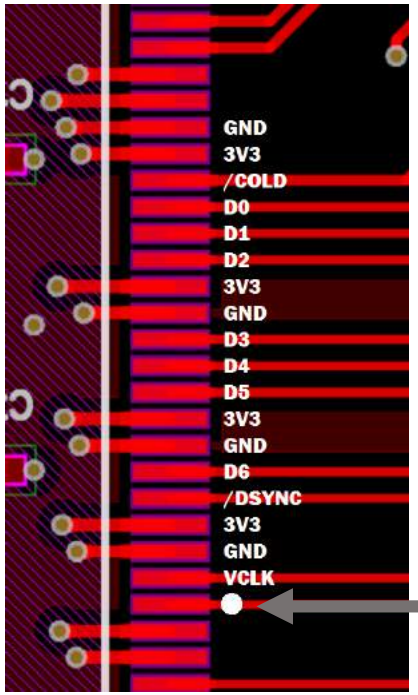
(no pictures atm)

- The N64RGB does not have a mounting hole to keep the size as small as possible
- Isolate the PCB from the bottom
- Fix the PCB into the heat sink (similar position as the N64A) using double side tape or other adhesive pads or hot glue

Solder Work – Console

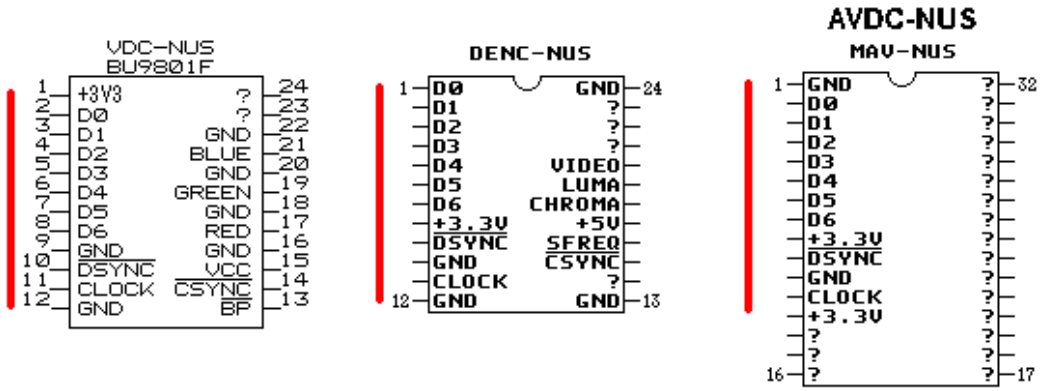
-- digital video signals --

Most of the signals needed has to be taken from video processor output; the **RCP-NUS**.
Next to the RCP-NUS is the video encoder, where several types are used during the variety of N64 designs.
The pinouts are given below. Basically we need D0-D6, /DSYNC and /CLK (or VCLK or CLOCK) for the modding board.



Pin 10 (marked with a dot)

Pinout RCP-NUS

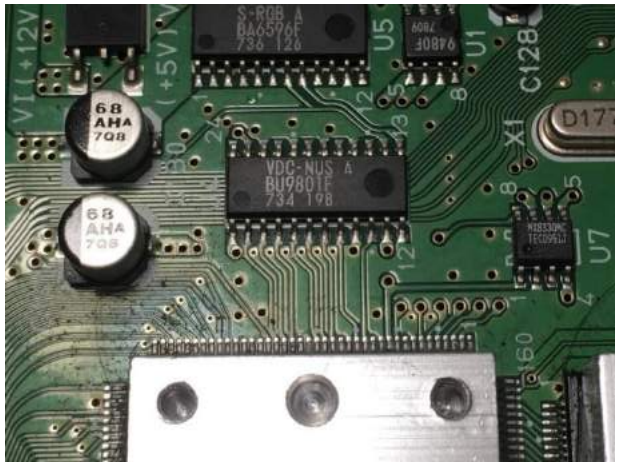


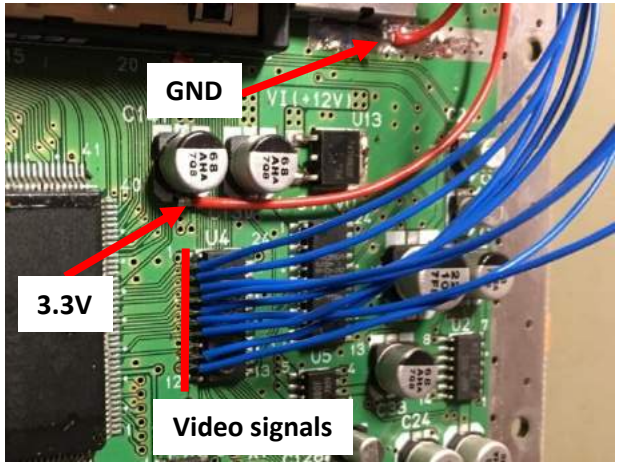
Pinouts of different encoder chips used in the N64

RCP-NUS picture by Marshall
Encoder picture by Viletim

Solder Work – Console

-- digital video signals and 3.3V and GND --

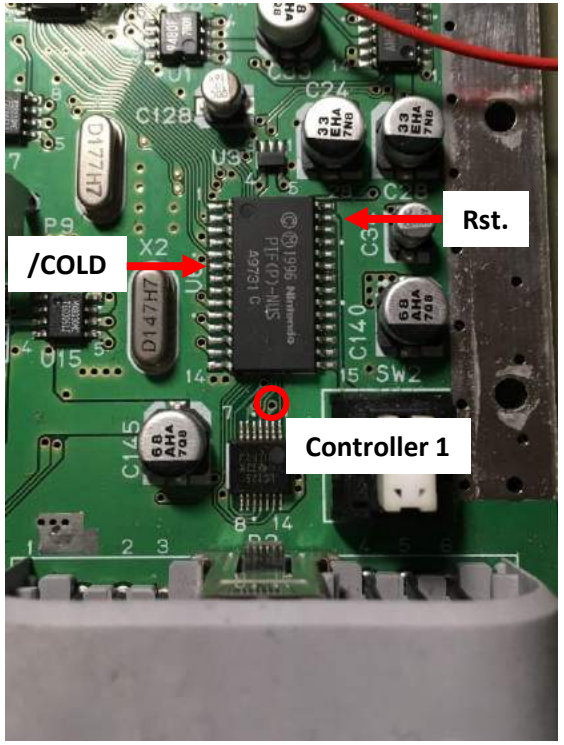


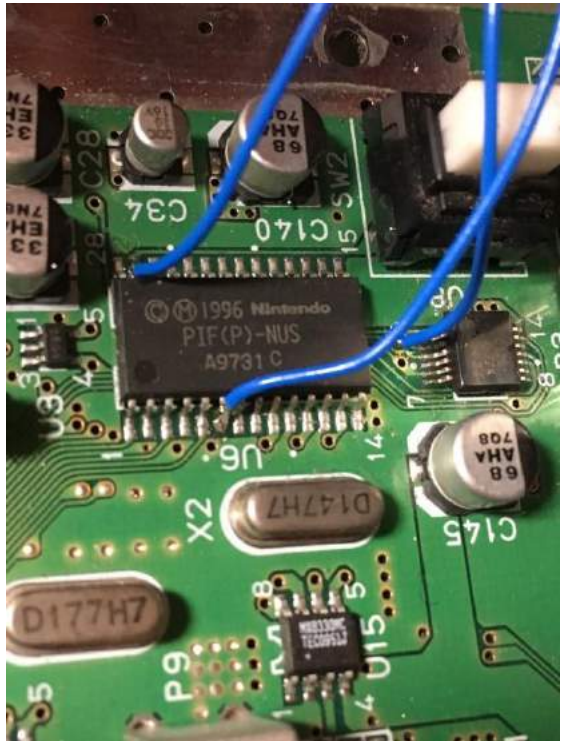


- Locate the video encode (here a VDC-NUS)
- Solder a bunch of wires to the video signal pins
  - You may use an adapter if you have a MAV-NUS or AVDC-NUS encoder (see README)
- Solder two addional wires
  one for 3.3V (can be picked off C141) and
  one for GND (large GND plain)
- Route all wires approximately to the MultiAV port

Solder Work – Console

-- Controller and Reset --

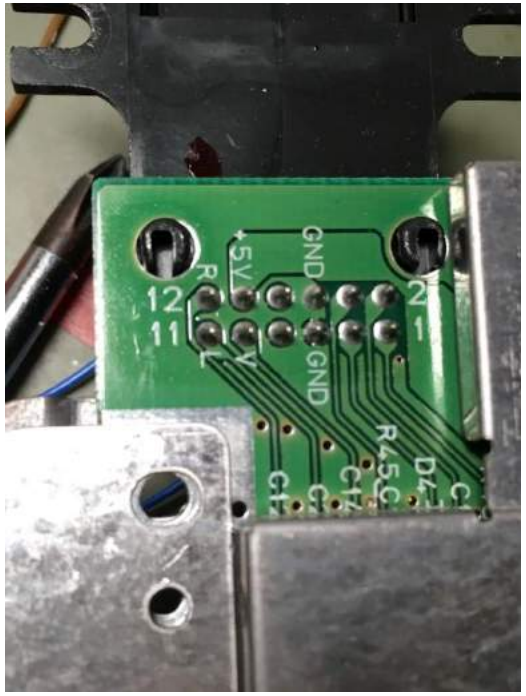


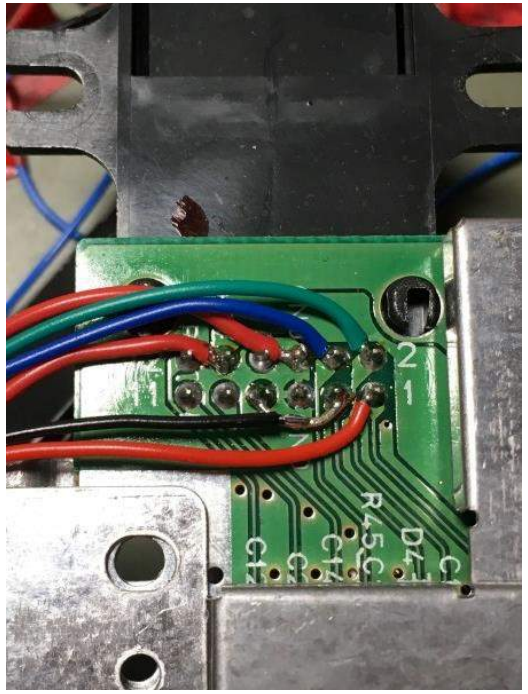


- All signals for the controller and the reset can be picked off the PIF-NUS which is located next to the reset button
- Solder the following wires:
  - One for reset (PIF-NUS pin 27)
  - One for the controller input
    in my case: the marked via, sometimes also PIF-NUS pin 16
    in any case: search for a suiteable connection to the middle pin of controller port 1
  - **N64A only:**
    one for /COLD (PIF-NUS pin 6)
    (atm not used in the current implementation)

Solder Work – Console

-- MultiAV port – analoge video output --



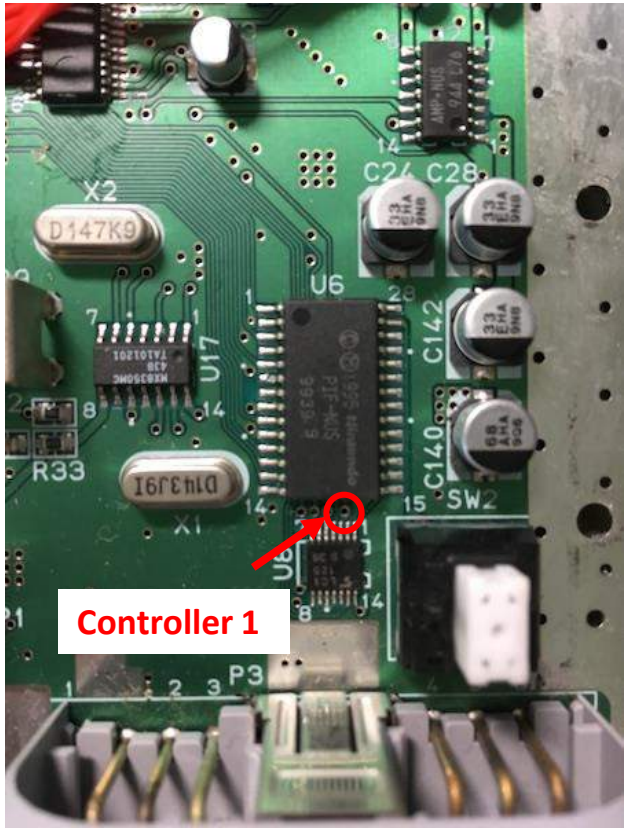


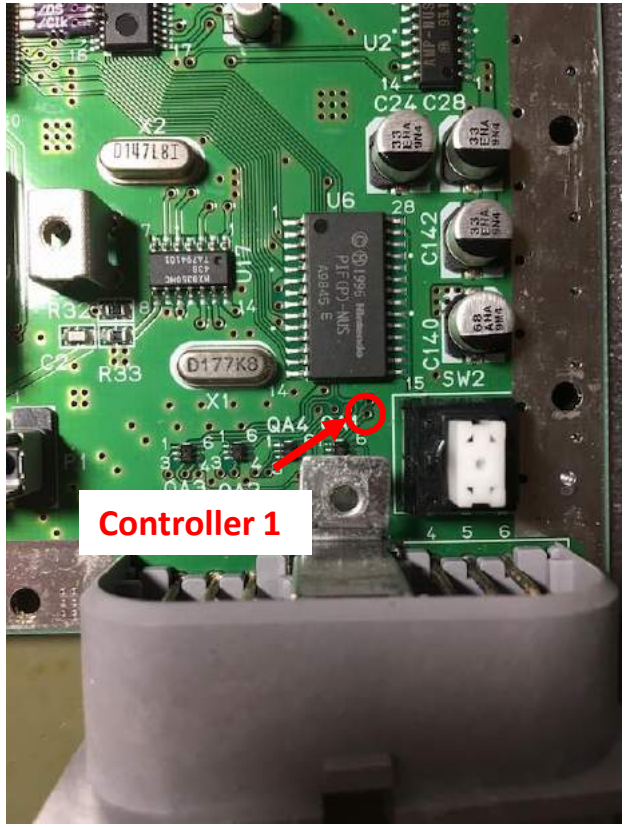
- Prepare some wires for:
  - Pin 1: Red / Pr
  - Pin 2: Green / Y
  - Pin 3: Sync *
  - Pin 4: Blue / Pb
  - Pin 5 / 6: GND
  - Pin 7: Sync *
  - Pin 10: +5V (N64RGBv2 and N64A only)

*** Sync:**
You need to take the composite sync out of the modding board. Depending on your RGB cable, you need to use pin 3 (sync on csync cable) or pin 7 (sync on luma cable). I soldered a bridge between both pins to make it useable with both types of cables.
Sometimes pin 3 / 7 is used in your stock console. **Make sure that the pin / these pins is / are unconnected.**

Solder Work – Miscellaneous

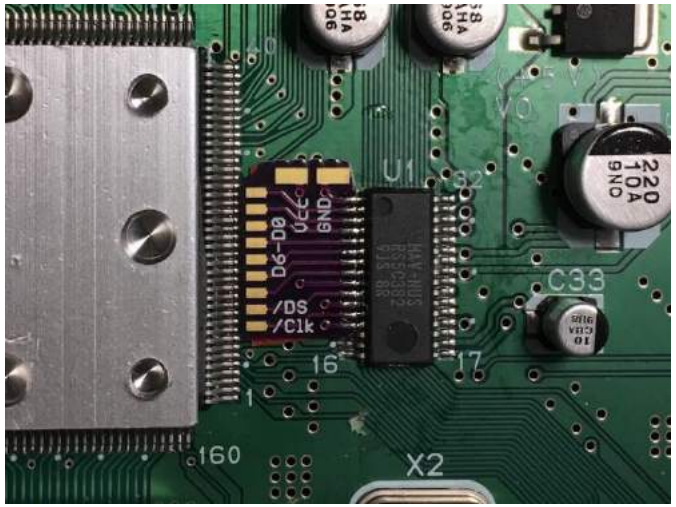
-- Other PIF-NUS configurations and proper Controller 1 solder points --





Solder Work – Miscellaneous

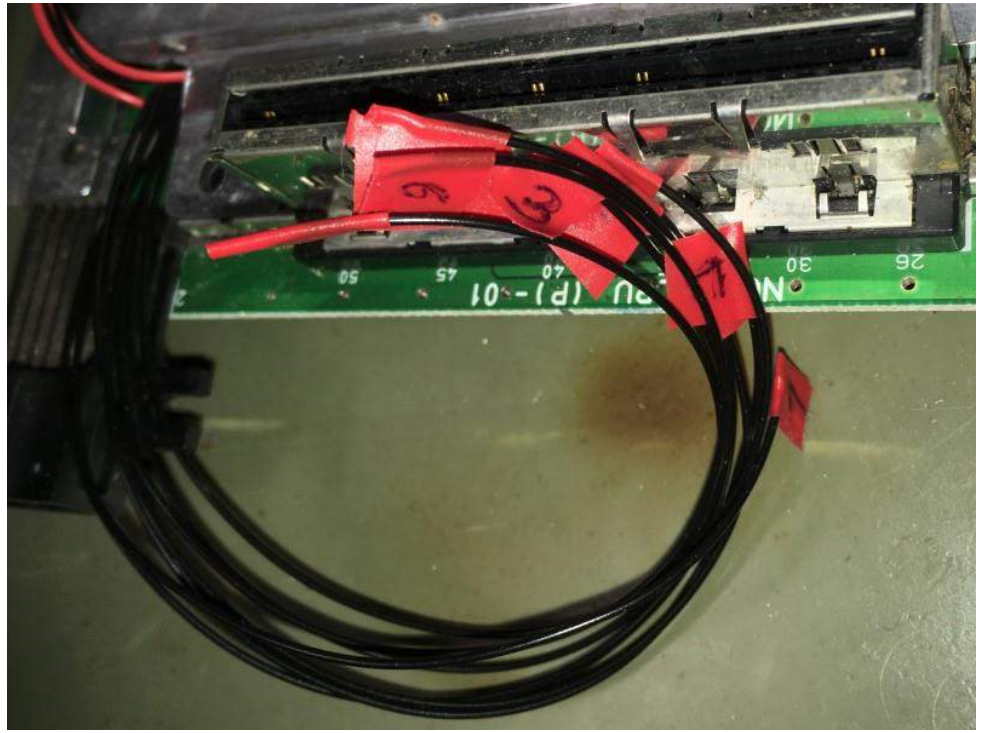
-- MAV-NUS--



- The MAV-NUS and the AVDC-NUS have a small pitch of 0.8mm
- You may want to use the MAV-NUS breakout board to work on a 1.27mm pitch
- Check connectivity between the RCP-NUS and the pads on the breakout with a DMM
- Check for adjacent shorts

Solder Work – Miscellaneous

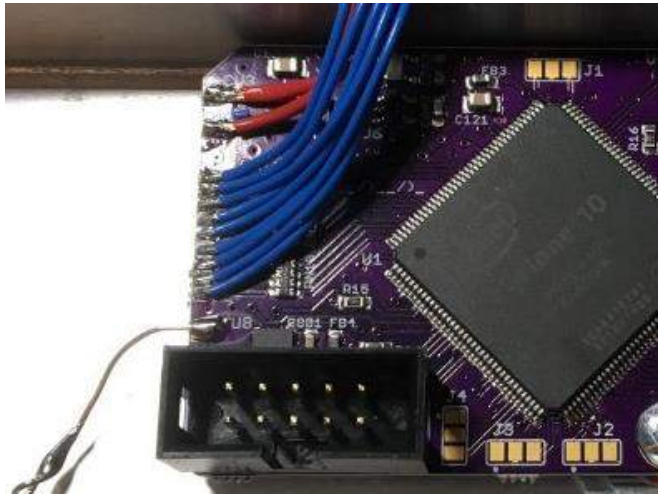
-- Marking wires --

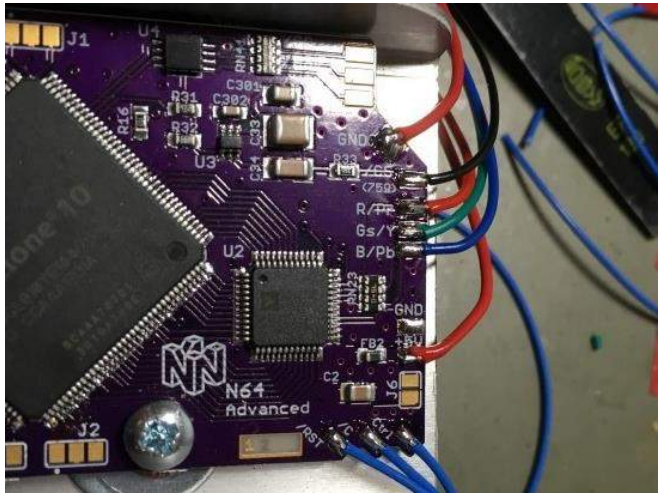


- If you use lose wires as I do, you may mark them before you go on

Solder Work – Modding Kit

-- N64 Advanced --





- Reassemble the top RF shield and heat sink on top of the N64 mainboard
  - Make sure that you do not squeeze any of the wires you prepared
  - For that you cut and/or bend the taps of the RF shield
- Connect +3.3V and GND to the left hand side of the modding kit
- Connect the seven data lines and /DSYNC and /CLK to the appropriate pads
- Connect the wires for /Rst., Ctrl. (Controller) and /C (/Cold) to the pads at the bottom right of the modding kit
- Connect the video lines, GND and +5V to the right hand side of the modding kit

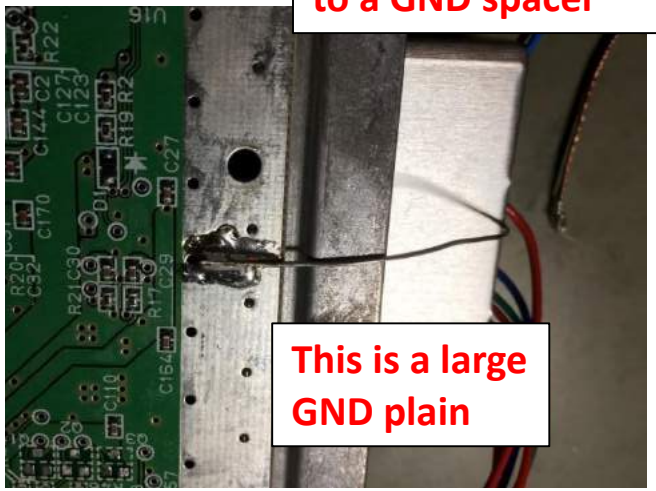
Solder Work – Modding Kit

-- N64RGB --

- Reassemble the top RF shield and heat sink on top of the N64 mainboard
  - Make sure that you do not squeeze any of the wires you prepared
  - For that you cut and/or bend the taps of the RF shield
- Connect +3.3V (Vcc) and GND to the left hand side of the modding kit
- Connect the seven data lines and /DSYNC and /CLK to the appropriate pads
- Connect the wires for /Rst. and Ctrl. (Controller) to the pads at the bottom right of the modding kit
- Connect the video lines and GND to the right hand side of the modding kit
- You may want to use the Fil pad to toggle the LPF on and off (see jumper section)

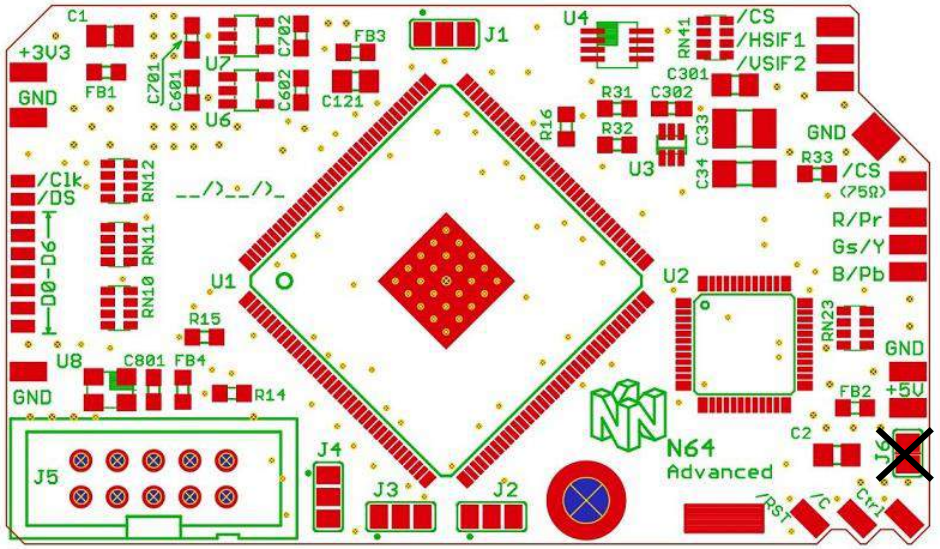
Solder Work – Modding Kit

-- Some extra GND --



- To reduce the GND loop size, it is helpful to assemble some extra GND connections to the N64 mainboard
- If the GND is not properly designed between the N64 mainboard and modding kit some current out of the digital signals flows back to the N64 throgh the analog video connections. This causes visible noise!
- I assembled two extra connection: one on the left hand side of the modding PCB and one at the right hand side.

Settings – Solder Jumper (N64 Advanced)



J5: 10pin JTAG interface
for update firmware

J1: VGA sync / Filter AddOn
J1.1: - opened: output HS, VS and CS
(VGA output possible)
- closed: use filter addon
J1.2: - Use filter board in bypass mode
(e.g. to use it as a simple
NTSC output to PAL output conversion)

J6: Analog power supply (outdated)
- leave J6 untouched and open!!!
Use 5V power supply for analog part

J4: Linemode
J4.1: - opened: linedoubling enabled
- closed: no linedoubling (beats J4.2)
J4.2: - opened: 480i pass through
- closed: 480i de-interlace (bob)

J3: Scanline strenght
(J3.2/J3.1) =
- (opened/opened): 0%
- (opened/closed): 25%
- (closed/opened): 50%
- (closed / closed): 100%

J2: RGB, RGsB, YPbPr
J2.1: - opened: RGB output
- closed: RGsB output
J2.2: - opened: RGB / RGsB output
- closed: YPbPr output (beats J2.1)

Fallback Mode:
- Fallback is active if reset button is pressed on powering on
- fallback: 240p/480i and RGB output

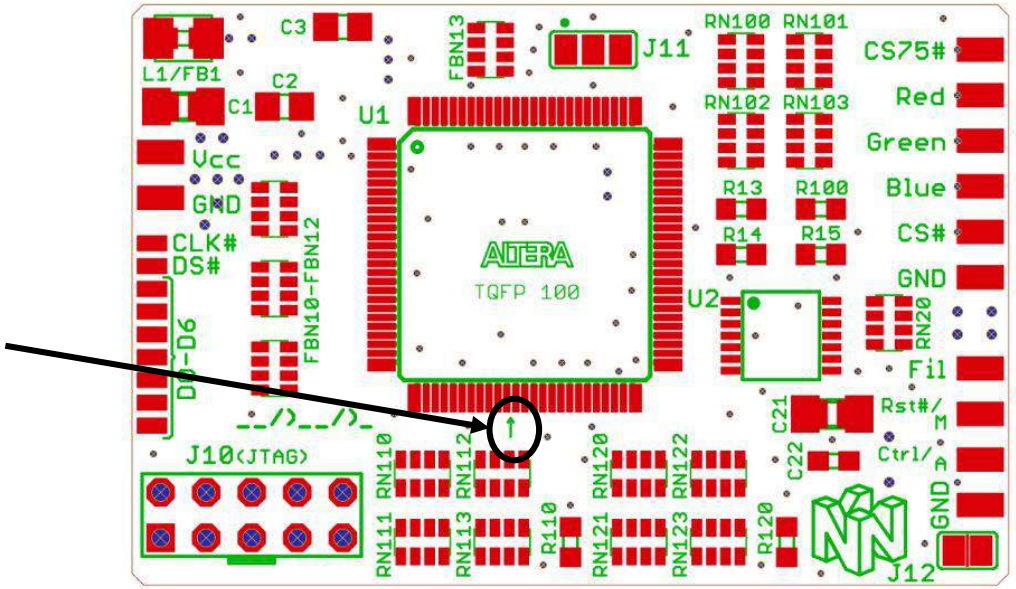
Notes for dual jumper:
- Jx.1 is always marked with a dot
- Closed reference is middle pad

Settings – Solder Jumper (N64 RGBv1)

J1: (bottom side)
- opened: use a CPLD with 240LEs
- closed: use a CPLD with 570LEs

Hidden Jumper between pin 36 and 37:
- short both pins to activate 15bit mode by default

J10: 10pin JTAG interface for update firmware



J12: IGR / Mechanical Switches
- opened: Use IGR (Ctrl. and /Rst)
- closed: Use switches at M and A pads

J11: De-Blur with IGR active

J11.1: (only applicable with heuristic off)
- opened: default de-blur on
- closed: default de-blur off

J11.2: - opened: use de-blur heuristic
- closed: bypass de-blur heuristic

Fil: Video filter
- opened: Use filter of the THS7374
- grounded: THS7374 in bypass mode

M: Manual (only with J12 closed)
- opened: no de-blur or de-blur heuristic
- grounded: de-blur in any 240p case (beats heuristic)

A: Auto (only with J12 closed)
- opened: bypass de-blur heuristic
- grounded: use de-blur heuristic

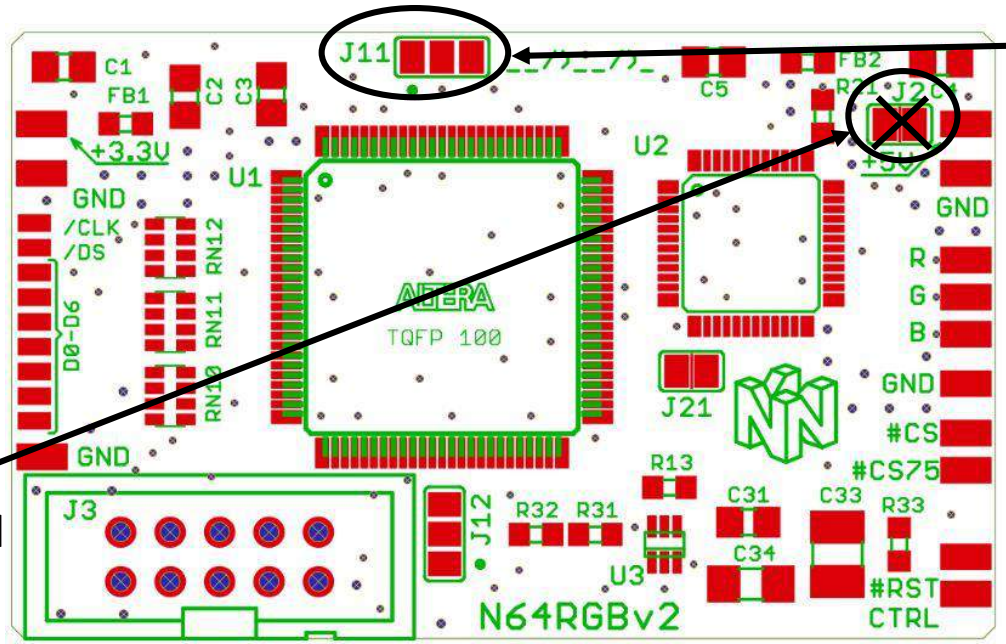
Notes for dual jumper J11:
- J11.1 is marked with a dot
- Closed reference is middle pad

Settings – Solder Jumper (N64 RGBv2)

J1: (bottom side)
- opened: use a CPLD with 240LEs
- closed: use a CPLD with 570LEs

J2: (outdated and formerly J1)
- leave J6 untouched and open!!! Use 5V power supply for analog part

J3: 10pin JTAG interface for update firmware



J11: De-Blur with IGR active
J11.1: (only applicable with heuristic off)
- opened: default de-blur on
- closed: default de-blur off

J11.2:
- opened: use de-blur heuristic
- closed: bypass de-blur heuristic

J21: Sync on Green
- opened: Don‘t use sync on green
- closed: Output sync on green

J12.2: IGR / Mechanical Switches
- opened: Use IGR (Ctrl. and /Rst)
- closed: Use switches at M and A pads

J12.1: 15bit mode default
- opened: 15 bit mode off by default
- closed: 15 bit mode on by default

M: Manual (only with J12.1 closed)
(#RST pad used for M)
- opened: no de-blur or de-blur heuristic
- grounded: de-blur in any 240p case (beats heuristic)

A: Auto (only with J12.1 closed)
(CTRL pad used for A)
- opened: bypass de-blur heuristic
- grounded: use de-blur heuristic

Notes for dual jumper J11:
- J1#.1 is marked with a dot
- Closed reference is middle pad

Cable Setup

-- N64RGBv2 and N64 Advanced --

Name	MultiAV	SCART plug	Ref. GND in SCART	Cinch plugs	Notes
Red / Pr	1	15	13	Red plug	Using a 220uF cap in series is possible
Green / Y	2	11	9	Green plug	Using a 220uF cap in series is possible
Blue / Pb	4	7	5	Blue plug	Using a 220uF cap in series is possible
Sync	3 or 7	20	17	Not needed	Consider notes on next page
GND	5, 6	4, 5, 9, 13, 17, 18, 21		To outer ring of each plug	Pin 21 @ SCART plug is outer shield
+5V	10	16	18	Not needed	Use a 180ohm resistor in series
Audio left	11	6	4	Red plug	
Audio right	12	2	4	White plug	

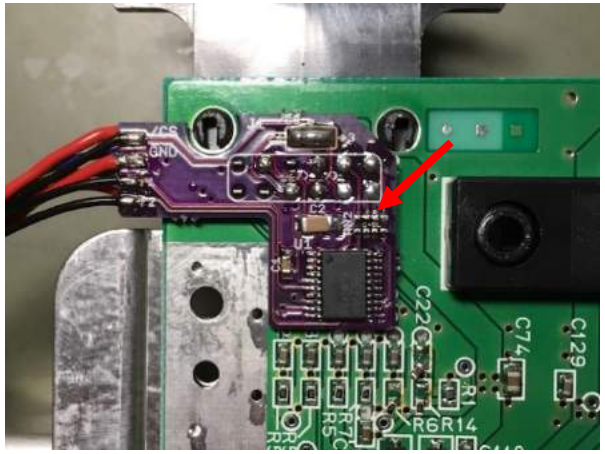
Cable Setup

-- N64RGBv2 and N64 Advanced --

- Notes on Sync:
  - **I recommend using the 75ohm compatible csync output**.
  - If you do so then run a straight wire through your cable for sync.
  - If you use TTL sync output, you have to add an additional resistor in series (N64RGv2: 0ohm to 900ohm; N64A: 270ohm to 1.1kohm) to attenuate the signal for 75ohm terminationed destinations.
- RGB cables:
  - If you buy an RGB cable, buy a typical RGB cable for a NTSC SNES with sync on luma (MultiAV pin 7) or sync on csync (MultiAV pin 3).
  - If you bought a raw csync cable and if you use the 75ohm output, remove / short the resistor which is possibly installed in the sync trace.
- RGsB / YPbPr Cables:
  - With the given table you should be able to build your own component cable or to build an adapter (see pictures on left side)
  - Unfortunately, there are no cables / adapter to bought.

Cable Setup

-- Filter Add On for the N64RGBv2 and N64 Advanced --



- Use the Filter AddOn if your ADC device does not apply input filtering (e.g. some capture cards)
- Solder the board onto the MultiAV and decide where /CS has to go – pin 3 and/or pin 7 – depending on your setup
- All connections to the N64RGBv2 and N64A are labeled
- A note on PAL SNES RGB cables:
  - By default the N64RGBv2 and N64A are not suitable to use it with PAL SNES RGB cables
  - By using the Filter AddOn board one can convert the output such that it is possible to buy a stock PAL SNES RGB cable (with sync on luma)
  - Just use the Filter Addon board and use a 39Ohm resistor array at the output (marked with a red arrow)
- N64RGBv2
  - To use the LPF of the adapter board, left F1 and F2 floating or connect them to GND
  - To bypass the LPF, connect F1 and F2 pads to +5V.
- N64 Advanced:
  - Remind to set J1.1 on the N64A
  - Set J1.2 if you want to have the filter off, e.g. in the case you simply want to use a PAL SNES RGB cable
    (actually they will be set to 95MHz cut-off, which is way above the video content)

Cable Setup

-- N64 RGBv1 --

Name	MultiAV	SCART plug	Ref. GND in SCART	Notes	PAL assembly	NTSC assembly
Red	1	15	13		Use 75ohm resistor to reference GND	Using a 220uF cap in series is possible
Green	2	11	9		Use 75ohm resistor to reference GND	Using a 220uF cap in series is possible
Blue	4	7	5		Use 75ohm resistor to reference GND	Using a 220uF cap in series is possible
Sync	3 or 7	20	17	Consider notes on next page	Using a 220uF cap in series is possible	
GND	5, 6	4, 5, 9, 13, 17, 18, 21		Pin 21 @ SCART plug is outer shield		
+5V	10	16	18	Use a 180ohm resistor in series		
Audio left	11	6	4			
Audio right	12	2	4			

Cable Setup

-- N64 RGBv1 --

- PAL or NTSC assembly:
  - Depends on RN20
  - PAL assembled kits have 39ohm resistor array installed
  - NTSC assembled kits have 75ohm resistor array installed
- Notes on Sync:
  - **I recommend using the 75ohm compatible csync output**.
  - If you do so then run a straight wire through your cable for sync. For PAL assembly an optional resistor of 75ohm to reference GND can be installed.
  - If you use TTL sync output, you have to add an additional resistor in series.
    - R15 = 330ohm … series resistor of size 0ohm to 420ohm
    - R15 = 0ohm … series resistor of size 330ohm to 750ohm
- RGB cables:
  - It‘s possible to buy pre-build RGB cable with either PAL SNES or NTSC SNES setup.
  - Use a cable with sync on luma (MultiAV pin 7) or sync on csync (MultiAV pin 3).
  - If you bought a raw csync cable and if you use the 75ohm output, remove / short the resistor which is possibly installed in the sync trace.

Controller – In Game Routines

Basis for all button combinations

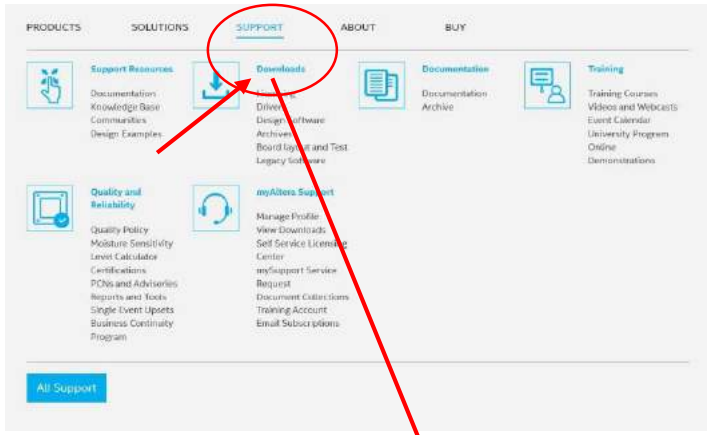
- The controller is able to trigger a reset or to change de-blur and 15bit mode option
- Button combinaton: **Start + R + Z + …**
  - **… A + B** : trigger a reset
  - **… C-ri** : de-blur for 240p on
  - **… C-le** : de-blur for 240p off
  - **… C-up** : 15bit mode off
  - **… C-dw** : 15bit mode on

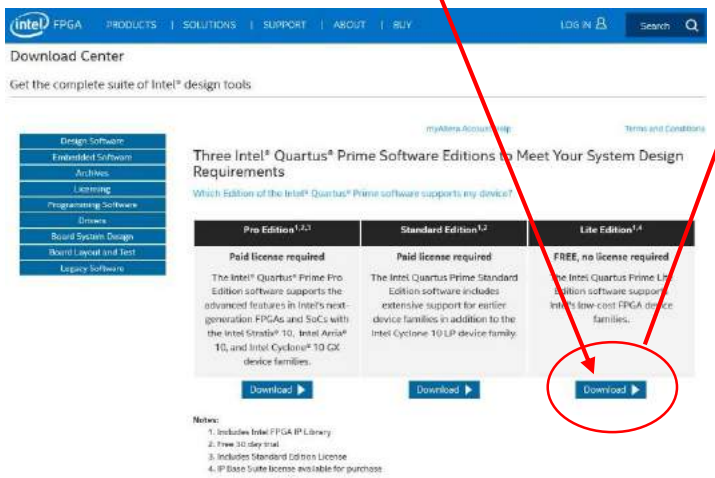
Notes:

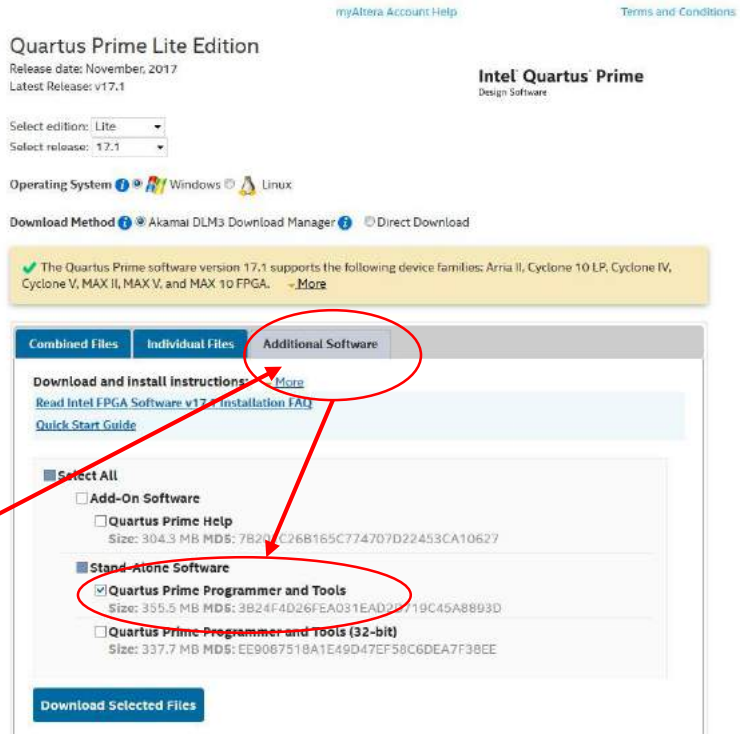
- Controller is only read if the actual game reads the controller
- Switching de-blur overwrites the heuristic estimation.
  De-blur option is reset with each triggered reset and power cycle
- 15bit mode is reset with each power cycle

Firmwareupdate

-- Download the software tools --







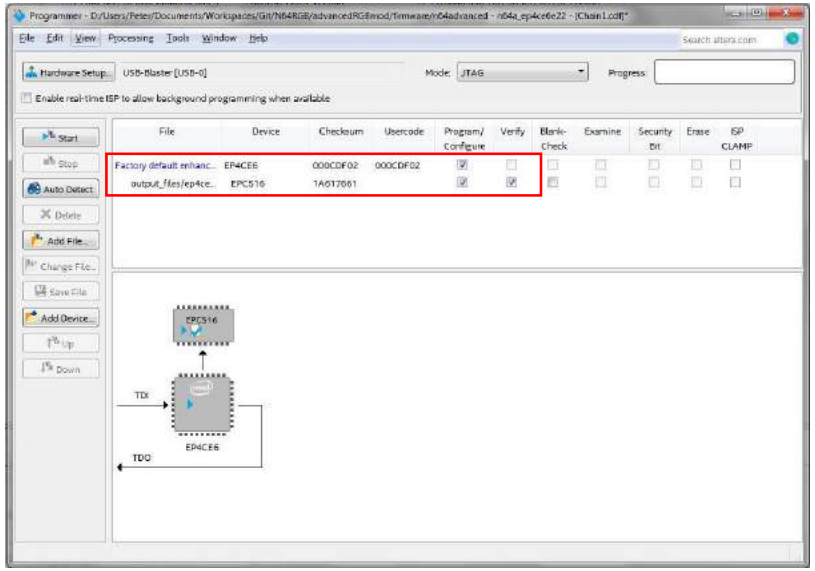
1. Go to altera.com and select “Downloads” from the “SUPPORT” menu
2. Select “Download” from the “Light Edition” column of the “Design Software” (preselected)
3. Go to “Additional Software”
4. Download the “Quartus Prime Programmer and Tools software

If you have the Quartus Prime edition installed, the programmer tools are already available on your computer.

Date: 22.11.2017
Website layout may change

Firmwareupdate

-- Update your N64 Advanced --

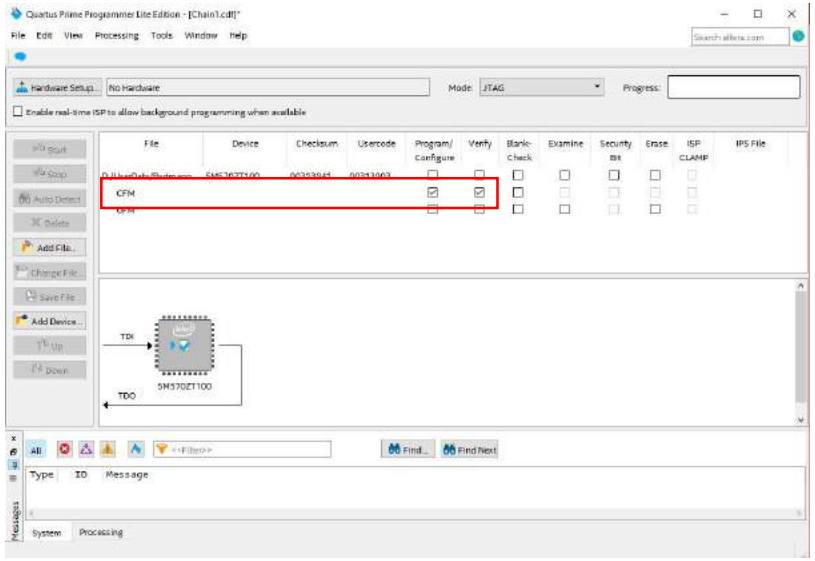


- Connect your PC with the USB Blaster and the USB Blaster with the modding kit J5
- Download the actual firmware build from the GitHub repository folder advancedRGBmod/firmware/output_files/*yourFPGA*/n64a_*yourfpga*.jic
- Open the programmer tool
- Load the previously downlaoded JIC file with “Add File…”
- Check “Program/Configure” for the FPGA and “Program/Configure” and “Verify” for the Flashdevice
- Make sure that the USB Blaster is selected in the “Hardware Setup”
- Click on “Start” (the N64 has to be powered for the reference voltage) and wait for the process to be finished
- Power cycle your N64 and enjoy :)

Quartus Prime Lite v17.1

Firmwareupdate

-- Update your N64 RGB --



- Connect your PC with the USB Blaster and the USB Blaster with the modding kit J10
- Download the actual firmware build from the GitHub repository folder generalRGBmod/firmware/output_files/v#/n64rgb_*yourcpld*.pof
- Open the programmer tool
- Load the previously downlaoded POF file with “Add File…”
- Check “Program/Configure” and “Verify” for the CFM part of your device
- Make sure that the USB Blaster is selected in the “Hardware Setup”
- Click on “Start” (the N64 has to be powered for the reference voltage) and wait for the process to be finished

Quartus Prime Lite v17.1