# Commodore Plus/4 PAL/NTSC switch

by Daniël Mantione <daniel.mantione@freepascal.org>

This document was written after I bought a Commodore Plus/4 from Ebay. As an unpleasant surprise, it was an NTSC machine. Allthough in 2018, it ain't that hard to find equipment that eats NTSC signals, in fact my big TV has no problem with it. But the Highscreen monitor that I use for my Commodores doesn't like it. So there is a desire for a PAL mod.

In a Youtube video, The 8-bit Guy did show how easy it is to modify TED based computers from PAL to NTSC. Allthough I did knew the TED could do both PAL and NTSC, this video did really show how easy it was and started me thinking. (https://www.youtube.com/watch?v=aXt6Obgm2G4)

So ot is easy to mod, but is it also easy to switch between PAL and NTSC? Having an NTSC machine around is useful for software development. PAL and NTSC incompatibilities were a bad habbit in the 80's that we can and should avoid in the 21th century.

I started Googling around and found this link:

http://www.fascinationsoftware.com/FS/html/Plus4-videoswitch.html

This guy did use a mechanical switch in the clock line to achieve the desired result. So it is possible, but I didn't really like his approach:

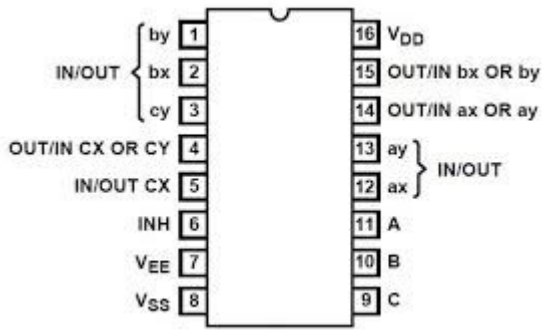
- The wires from/to his switch carry high frequency clock signals that are good radio sources, or good antenna's. Slight disruptions near the crystal can disrupt the clock generator, so I have quality doubts here. The oscillator is inside a Faraday cage for a reason.
- That mod is rather ugly, the machine is forever mutilated. He blames his Dremel skills, but that is is not the root problem at all. The main problem is that he uses a big switch, needs big holes in the case and very few people can make these perfectly. The large number of cables further detoriates the look.

So I started thinking about ways to do a “clean” mod. A mod that doesn't require high frequency clock signals on long wires and is optically nice and elegant for the machine. After some brainstorming, I got an idea around a analog multiplexer. Would it be possible to switch between PAL and NTSC clocks with an analog multiplexer? This would have a few advantages:

- It is possible to fit a CD4053 inside the Faraday cage, only a control wire needs to run to a switch
- An SPDT switch is sufficient, allowing the use of small, compact switches and reducing the amount of wires that run inside the case

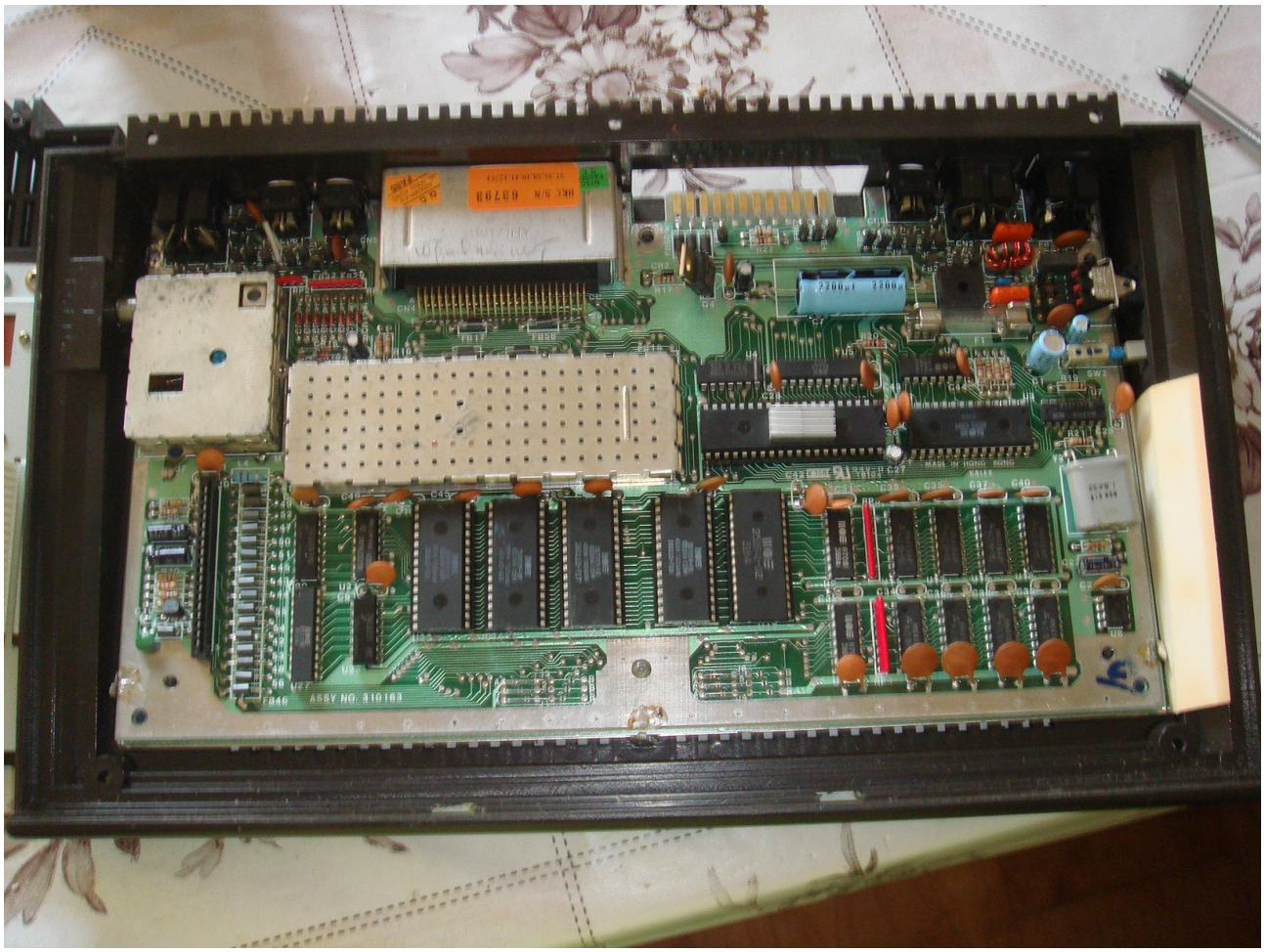
I ordered a few CD4053 plexers from AliExpress and started experimenting.

The CD4053 contains 3 analog multiplexers:



For example you connect PAL clock to ax, pin 12, NTSC clock to ay, pin 13, and then with the A pin 11 you can select which clock appears on pin 14. At least, that is the theory.

The machine before work starts. It had already received a CPU heatsink and a Commodore 64 style power connector earlier:

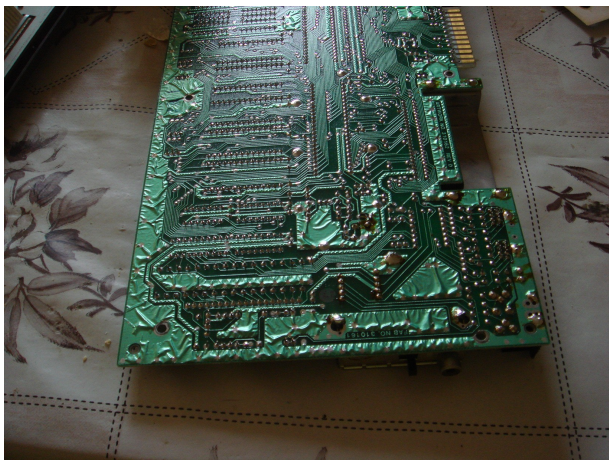


I started looking for good places to add a switch. Ideal would be a place inside the cartridge or user port, as this removes the need for drilling in the case. But I found no good possibilities here. Now you don't switch between PAL and NTSC daily, therefore you do not need a big switch in a prominent place. Instead, a small switch with a small hole in the case should be acceptable.

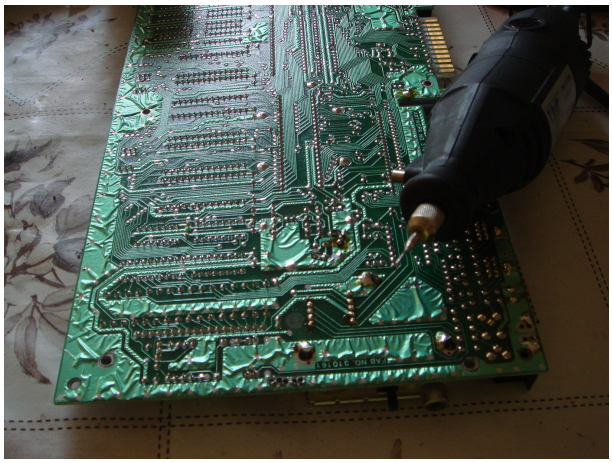
After carefull inspecting the board, I decided that the best place was indeed the same place as the other guy with his big switch did use: next to the modulator.

To avoid drilling extra mounting holes in the case, I decided that I wanted a PCB mounted switch. And that was the first step.

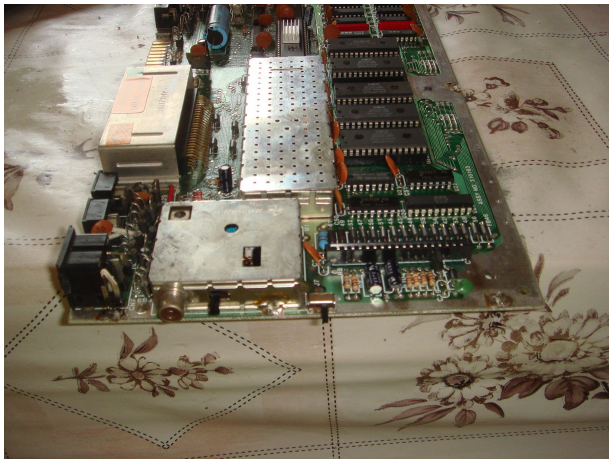
Back of the mainboard:



And putting my Dremel clone mini drill into action, I did drill 5 small holes into the mainboard:

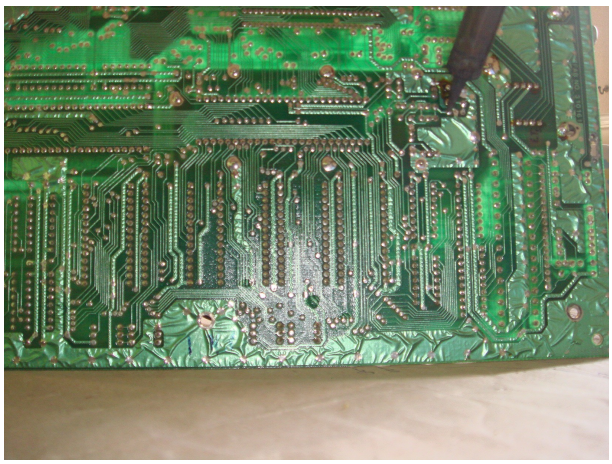


Testing if the switch fits:

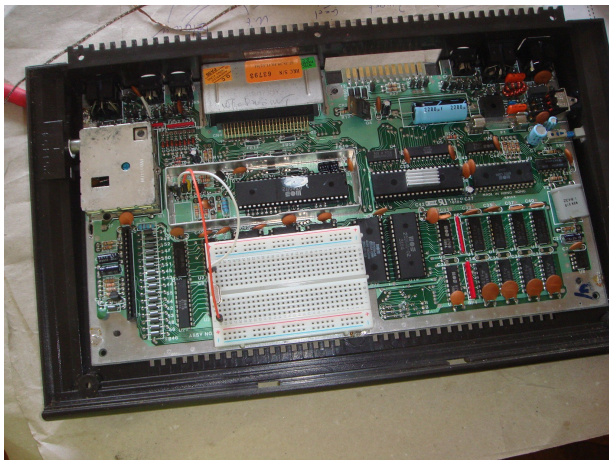


... perfect. And it does look nice too!

Next step was to unsolder the crystal:



... and I soldered some wires instead in order to be able to test on a breadboard. I put a 17.73 MHz crystal and ceramic capacitor on the breadboard and we could do the first test at PAL frequency:



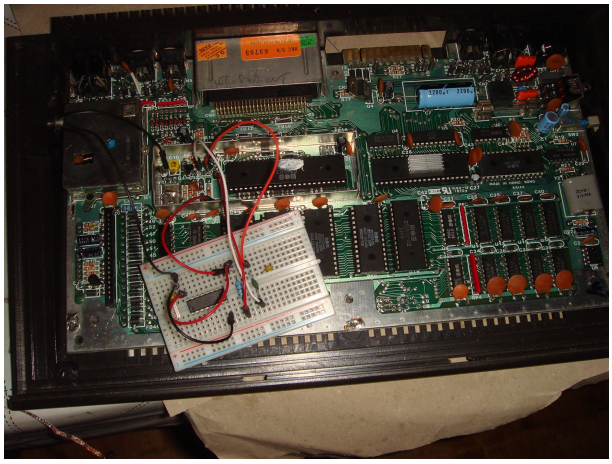
You modify the crystal and there still is a picture?!


Clever modern equipment! :) But the oscillator frequency was really 17.73MHz now:

Indeed I did use my living room TV for testing, as I figured that while developing the switch, I would need to test both PAL and NTSC, so using my Highscreen CRT monitor would not be ideal.

Time to start adding the CD4053 and some more components on the breadboard. The CD4053 needs 5V, which I got by unsoldering the inductor L3 from the mainboard and putting it on the breadboard.

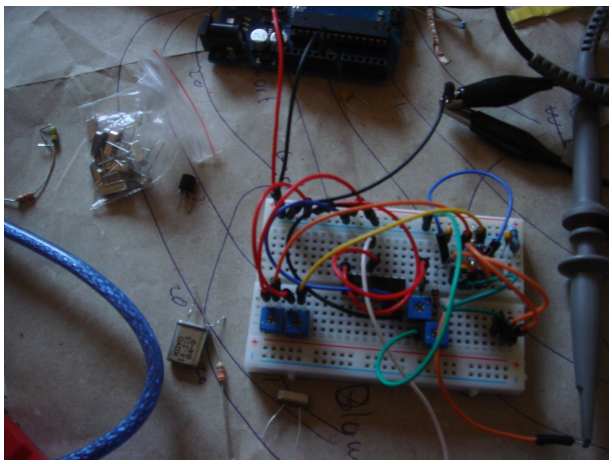
First attempt was to use the plexer to switch between both crystals:



This was way too optimistic. An analog multiplexer should ideally act like a mechanical switch but it isn't perfect. And the Collpitts oscillator on the board is quite sensitive: The plexer in between was enough to disrupt it and the result was a flat 0 Hz clock signal.

Therefore a more complicated approach was necessary. After some thinking, I decided to clone the Collpitts oscillator from the mainboard on my breadbord. The mainboard oscillator would always run at NTSC frequency, my clone on PAL frequency and the plexer would switch between both clocks.

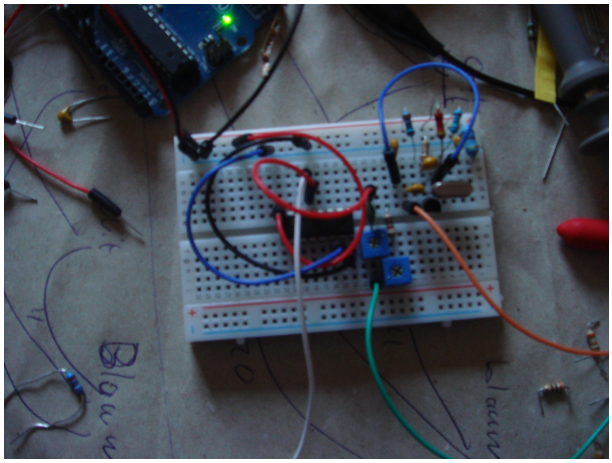
The amount of components are wires on the breadboard quickly grew:



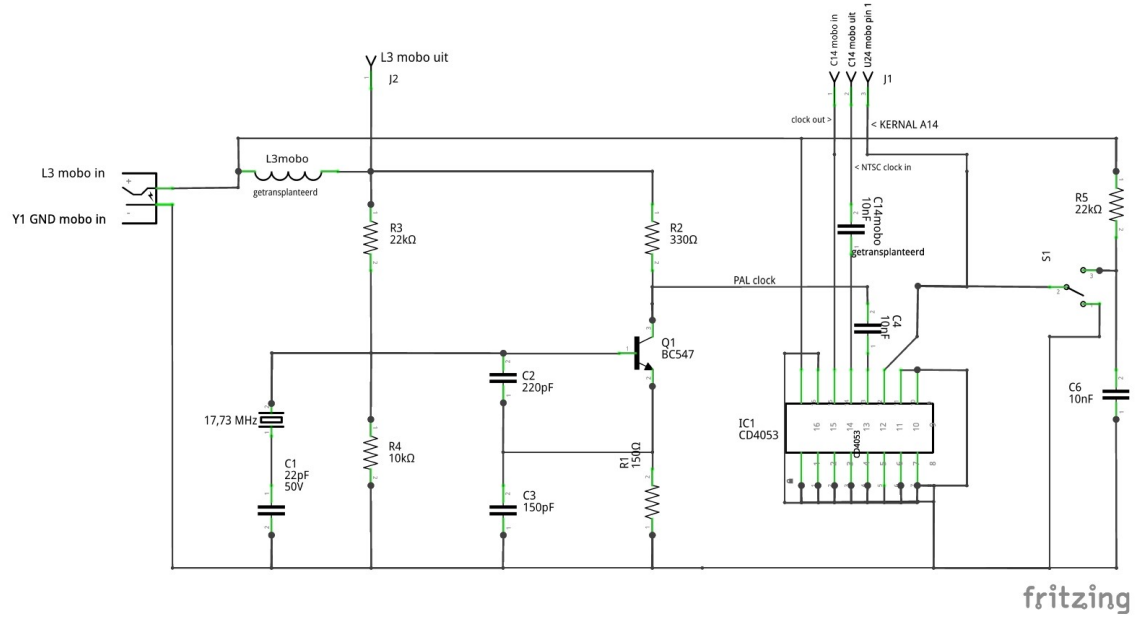
Building a clone of the oscillator was a bit harder than expected, I did not have exactly the same components, so improvised a bit and it initially didn't work at all. The main issue was related to the amplifying transistor, 2SC1815 on the mainboard. As I am in Europe, my box of components obviously doesn't contain many Japanese parts. Now, which transistor can be found in every component box in Europe? You guessed it correct, a BC547, so that is what I used. These Japanese transistors aren't difficult to find at all, but I saw no reason why a BC547 shouldn't work, because both are general purpose NPN transistors. Therefore I tinkered with it for a while, and as soon as I started to use potmeters instead of fixed resistors, I quickly got success when tuning the pots.

I spent quite a bit of time working on the amplitude of the clock signal, which was worse than the mainboard did generate. However, this turned to be unnecessary in the end, you simply get less amplitude at PAL than at NTSC frequency. In fact I got a better amplitude at PAL frequency than the original oscillator gave.

After some simplification steps the breadboard did look like this:

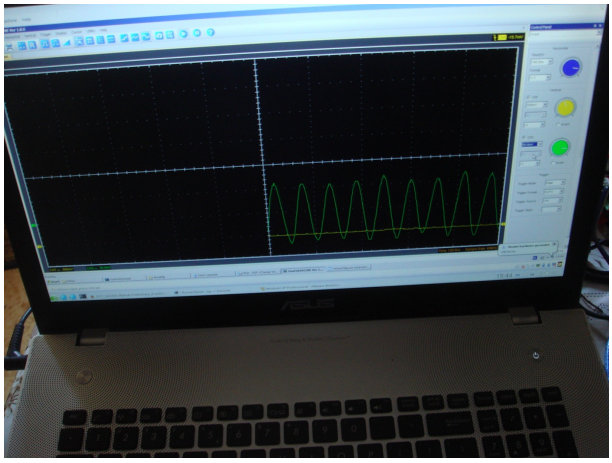


Time for a little schematic that shows the slightly different resistor values that I used in order to make the oscillator work optimally with the BC547:

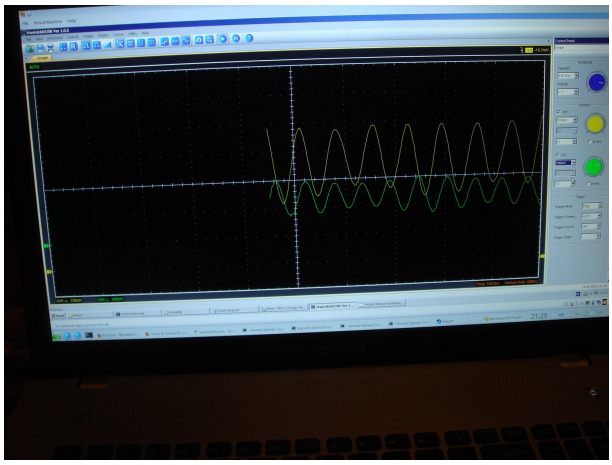



The schematic already shows the switch and the wire to the KERNAL that will be discussed further on.

The cloned oscillator signal on oscilloscope:



PAL clock (from breadboard) and NTSC clock (from mainboard) shown together:

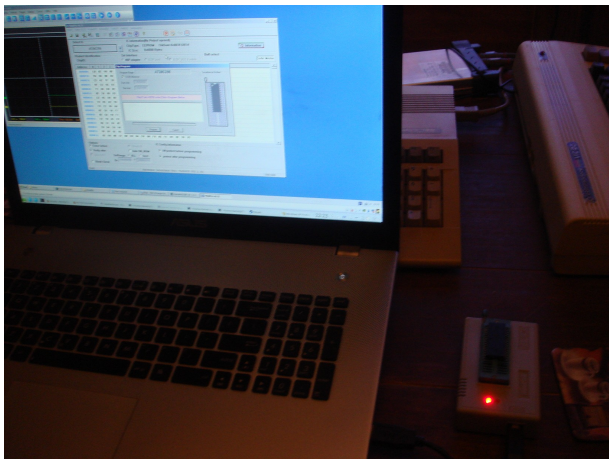


The NTSC clock certainly has better amplitude and looks a lot nicer, but the original oscillator doesn't do better at PAL frequencies and the Plus/4 has absolutely no problem with the lesser amplitude. But, this is the reason why I spent quite a bit of time on tuning it.

In order to get a machine output PAL, besides the right clock, a PAL KERNAL is needed. Time to

deal with that. The Plus/4 stores its KERNAL in a MOS 23128 16KB ROM. There are a few options for modern programmable ROMs. I have used a 32KB AT28C256 here, so the PAL and NTSC KERNALs can fit in a single ROM.

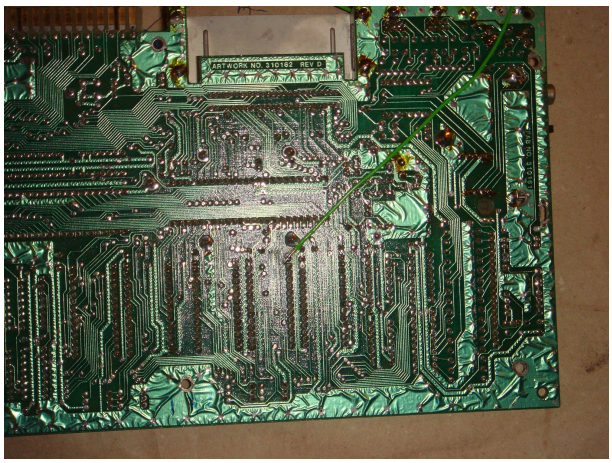
This ROM is a rather elegant choice, because it is pin compatible with the 23128. Really nice is that the extra address line needed for the 32KB capacity, A14, is on pin 1, which is NC for the 23128. And indeed, pin 1 of the KERNAL socket on the Plus/4 mainboard is not connected to anything. So we can just put an AT28C256 in the KERNAL socket, and connect pin 1 to the switch. Time to burn a ROM!



Also nice is that the AT28C256 is still being produced today, so rather than depleting old stocks, we'll help with the demand necessary to keep these chips in production.

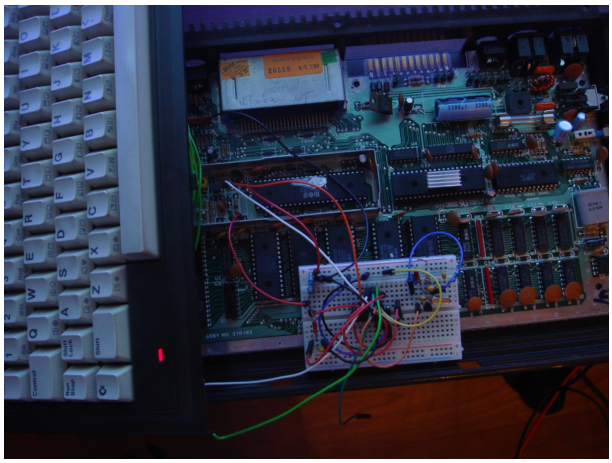
Rather than burning the standard KERNAL, I went with JiffyDOS. With the Commodore 64, allthough tapes are no longer part of daily use, I can imagine that I might ever want to read an old tape. For the Plus/4 though, I am 100% sure that I will never use a tape recorder in combination with it. JiffyDOS will be useful to connect it to my JD-equiped 1571, which should be quite a bit faster than even a 1551 drive.

I decided to solder a simple wire first to pin 1 of the KERNAL circuit in order to test the circuit:

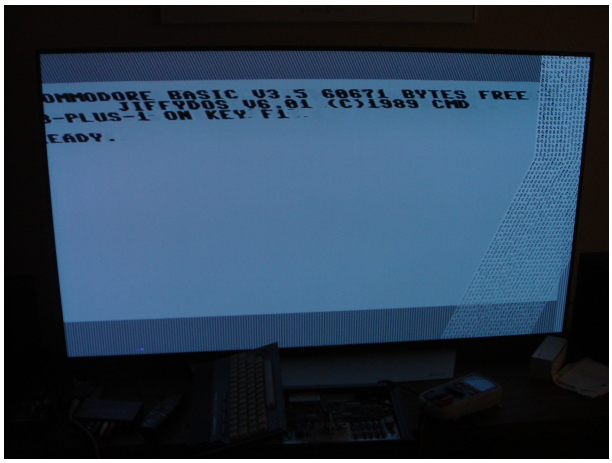


After connecting it to 5V the machine was ready to boot in PAL mode for the first time and it did!


The picture even looks slightly better than on NTSC. The machine in its current state:



A little attempt to run a PAL KERNAL at NTSC frequency. Unlike NTSC @ PAL frequency, the picture is no longer perfect:

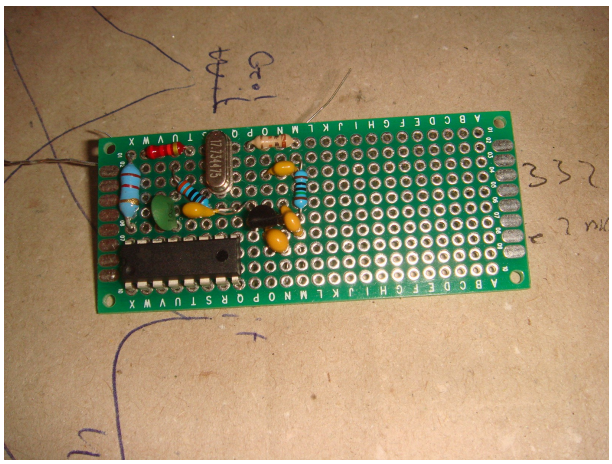



... but it is still impressive that there was a picture at all.

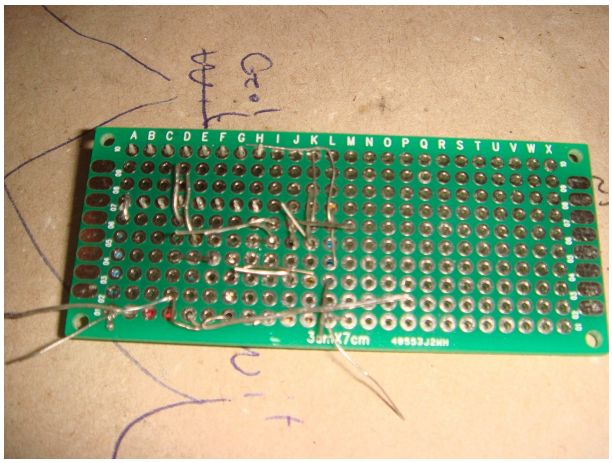
So, the concept was working! But lots of work did remain to integrate everything in the Plus/4 in a tidy way.

First, the circuit had to be transfered to a PCB. I decided that designing a custom PCB was overkill here and went with an experimentation PCB.

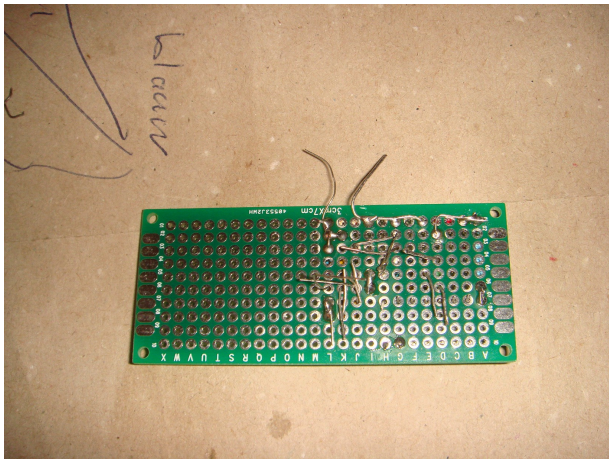
Arranging all the components on the PCB:



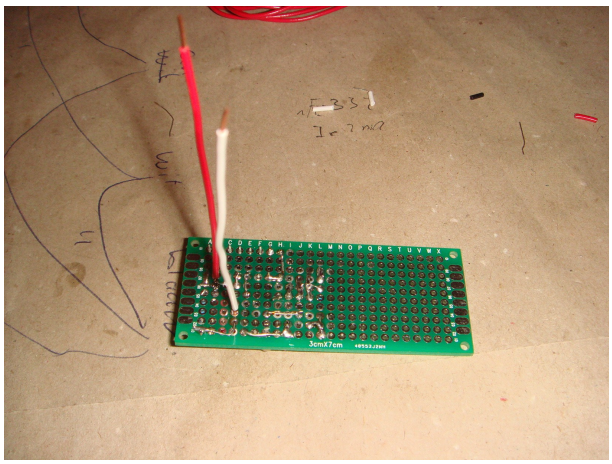
Big mess of component legs:



Partially soldered:

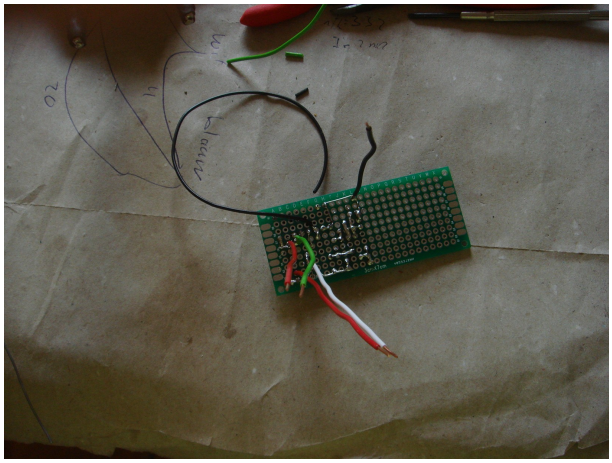


Fully soldered and first wires attached:

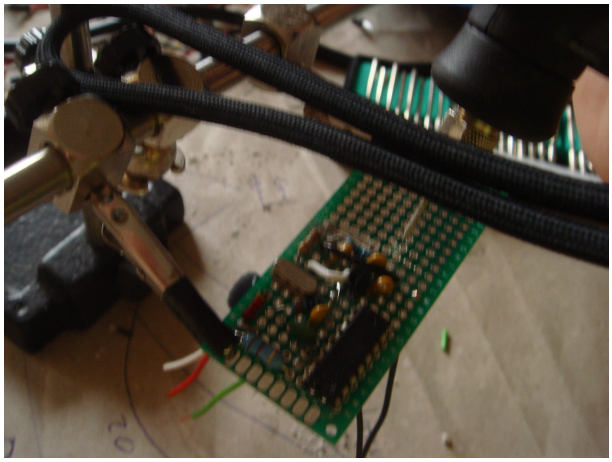


I decided to use solid core wires, because these will hold the PCB mechanically in place. This removes the need for a way to mount it.

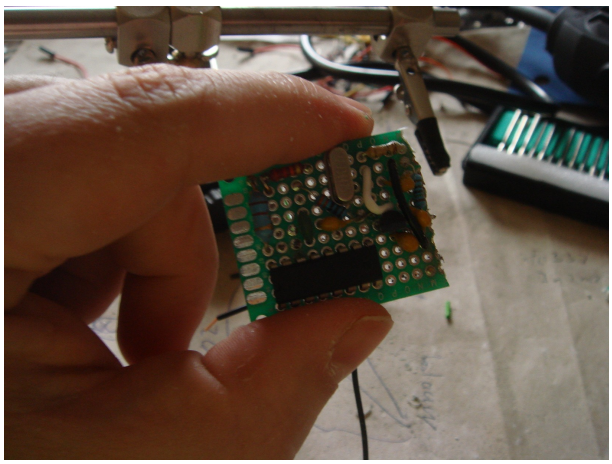
Ready for installation:



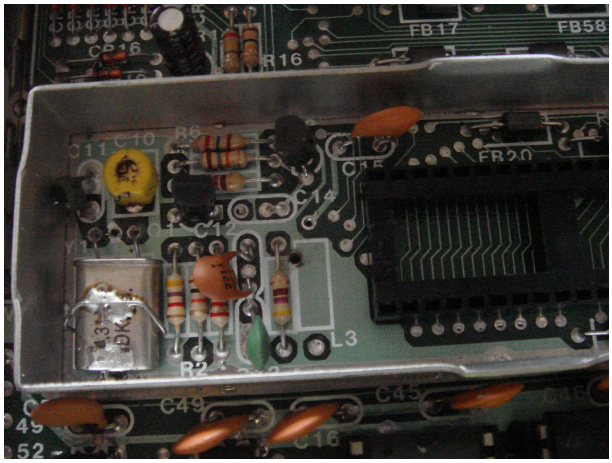
To fit into the cage, the empty space of the PCB had to be cut off:



A bit ugly, but I'll fix that later:

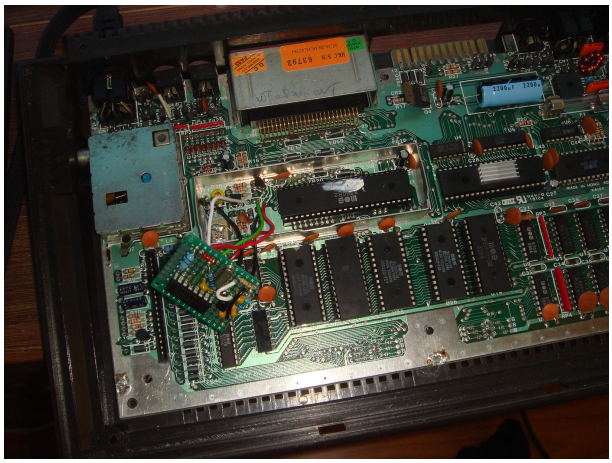


The PAL/NTSC switch is outside of the Faraday cage. Therefore there will be a wire from the switch that has to go in the cage. I decided that the best way to do this was to drill a small hole in the PCB. After selecting a good spot:



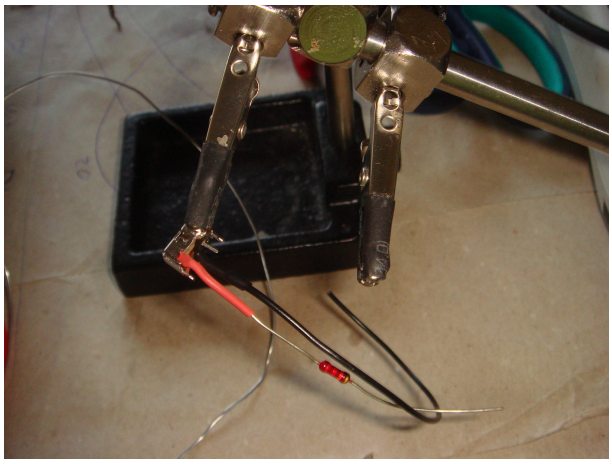

As you can see, I also put back the NTSC crystal and removed the wires. The PCB receives the NTSC clock at C14, which has been removed and is now on the PCB, and outputs either the PAL or NTSC clock at the other pin of C14. Obviously the TED and other chips were not on the mainboard during drilling.

PCB installed:

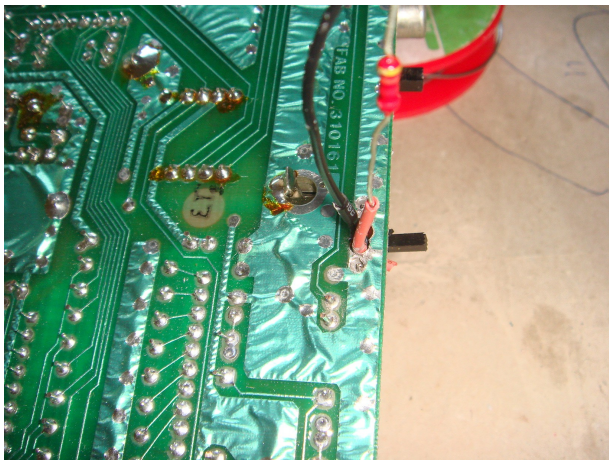


Time to have a look at the switch. Because the holes drilled in the mainboard go trough the ground plane, the pins of the switch are very close to ground. Proper insulation was necessary, so except for the pin connected to ground, the other legs were fully covered with heat shrink tube, which was cut in half at the top in order to be able to extend to the left and right side of the switch.  I also decided that using 5V directly had a too high risk of short circuit, do I decided to use a 22 kiloohm pullup

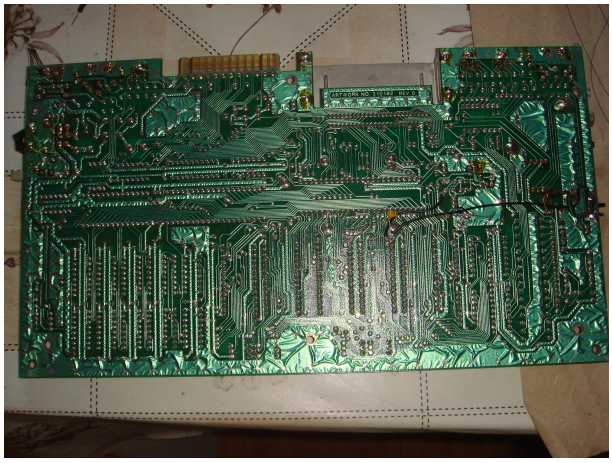
resistor:



Switch in final position:



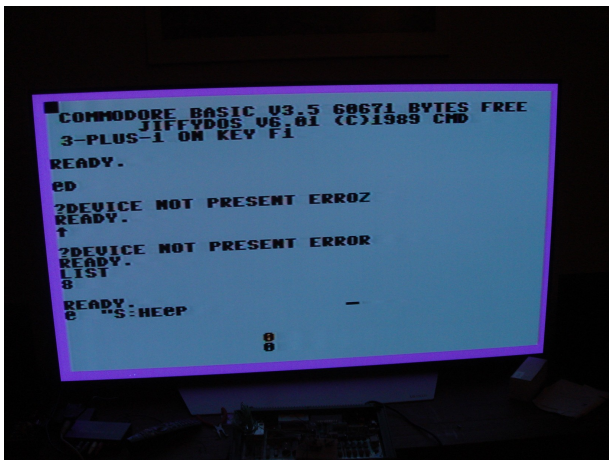
And the switch and wires soldered:



The wires on the underside of the mainboard are the part of the mod that I like the least, they are not held into place and therefore it looks not elegant. However, the underside of the board will normally not be visisble as it is covered by the RF shield.

You can see a ceramic capacitor in the above picture. This needs a bit more discussion. I noticed the system was unstable in NTSC mode, that was, with the switch connecting the pullup resistor.

There were automatic keypresses and random characters appearing (not instant, but did build up in several minutes):




After a bit of thinking, I developed a theory that IC switching inside the AT28C256 could cause the voltage on A14 to drop for very short times, and due to the 22kohm resistance, a small amount of time might be needed to get the voltage high enough again. As the CD4053 uses CMOS logic levels with 3.5V for high, just a little voltage drop is necessary for the CD4053 to start undesired switching between clocks.

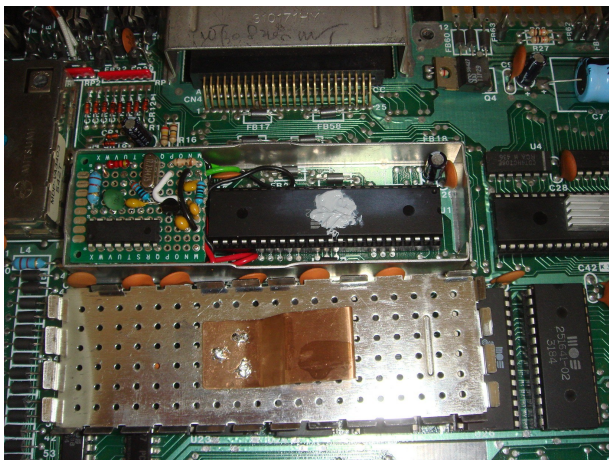
I reduced the value of the pull-up resistor and indeed, the instability decreased, but did not disappear. When 3.3 kOhm proved not to be enough to guarantee stability, I decided to add a 10 nF capacitor instead and this did eliminate the problem. I could go back to 22kOhm with no problem.

Final tests, PAL:


NTSC:

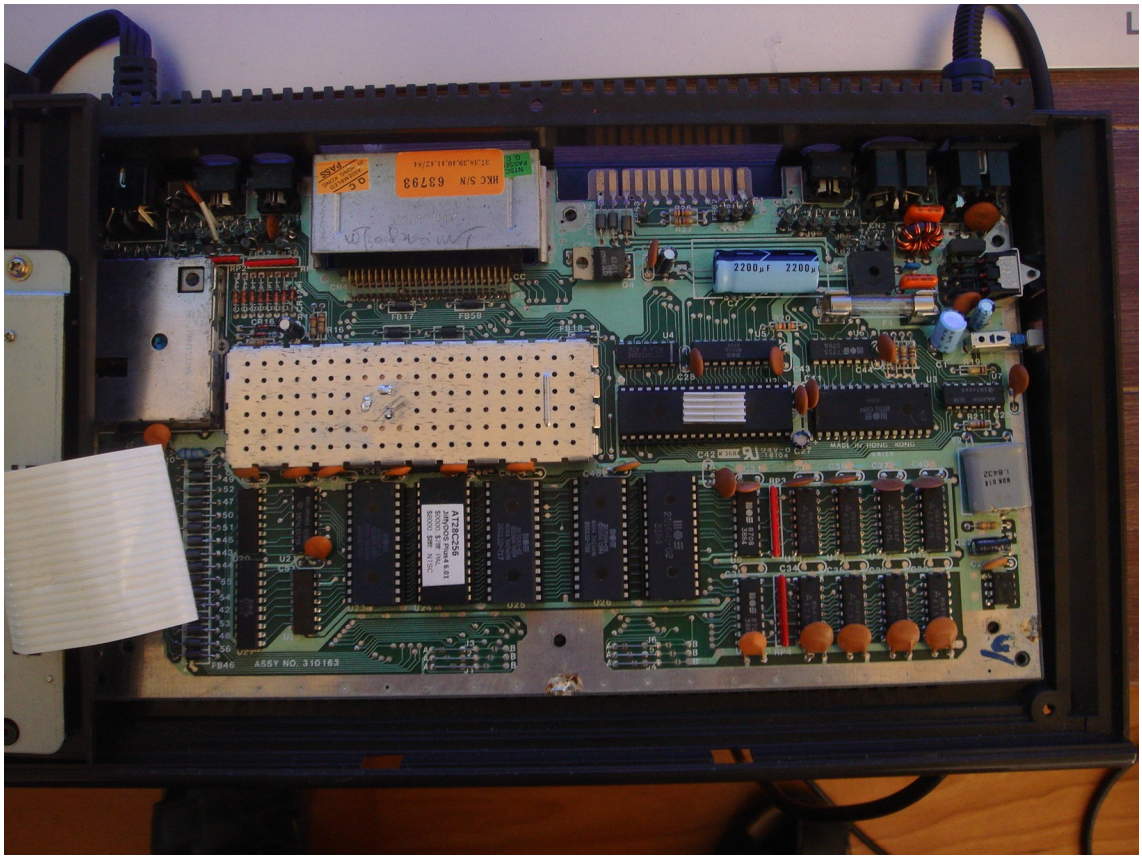

Works fine!

Time to push the PCB in place:



The TED got new thermal paste.

Board modifications ready! The end result:

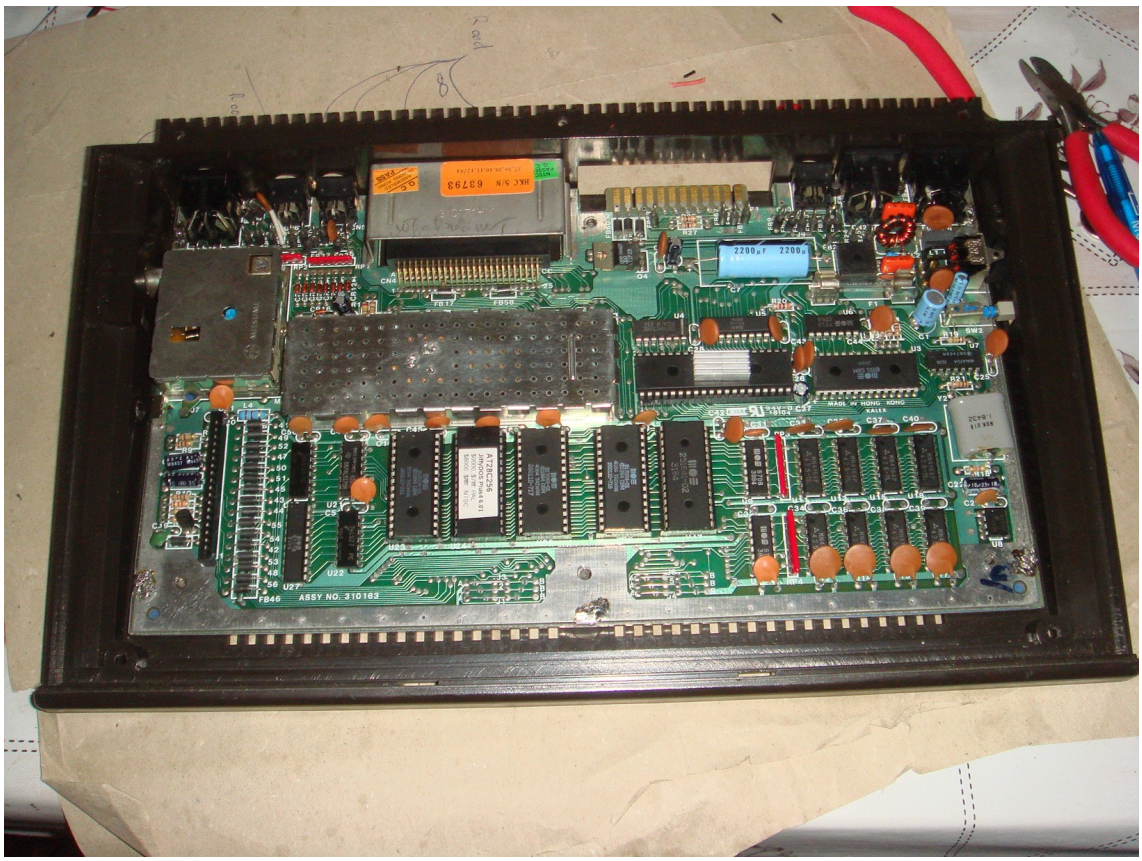


When looked at from the top, everything gives a factory-installed impression, including the PAL/NTSC switch, you cannot tell it was installed later without looking at the underside, for which you will have to remove the RF shield first. The AT28C256 EEPROM got a label to document what is inside.

One ugly job remains to be done. Drilling inside a classic computer should never be done without second though, but it was the best choice here. I made a small pilot hole with my mini drill, then used my big drill. 1-2-go-done.

End result:

Another look at the board:



Reassembled computer:

While it doesn't give the impression of a factory-installed hole, I am not unsatisfied with the result. It certainly doesn't look ugly and is perfectly functional. Compared to the mod in the article linked at the beginning, it is a significant improvement on both the optical and electronic aspects of the mod.

Now it is time to ban this machine from the living room! Using 320x200 on a 55 inch television is a terrible user experience, after all. The CRT in the hobby room is a much nicer solution for these old computers and the computer looks happy with the monitor: