A Pause button for the Atari 2600

February 20, 2010

Clique aqui para a versão em portugues

A major recurring subject in some forums that I participate always have been "How to pause an Atari game?". In a quick search, there are threads created in 2005! At least 5 years in search of a method to pause the game and the possibility to continue next. Although I have read the threads with curiosity and have seen some suggestions but I don't know much about.

Some time ago a colleague asked me to put a composite video output in an Onyx Jr. It's the only version of Atari with a pause button, so I took a good look to see how it works.

The first thing I noticed was that it used a 6502 processor and not the traditional 6507. The 07 is a version, let's say, "economic" of 02. Although they are compatible in the software, 07 has fewer number of signals, so fewer pins.

With some observations and measurements, I found that when the the pause button is triggered, two things happen:

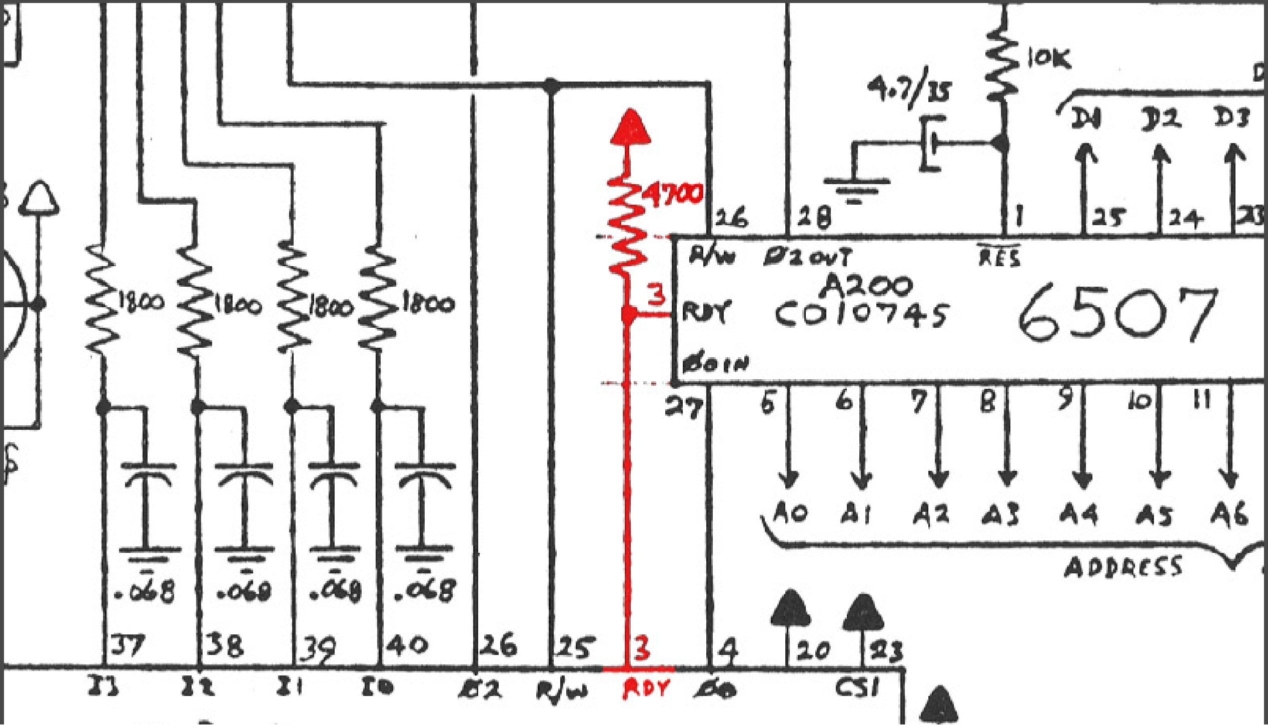
1 - The luminance signal and the audio signal are tied to ground, so the screen is black and has no sound. Without this we have some colorful patterns, just like a bad cartridge connection. This is because the image of the game can not be generated by the lack of processing.

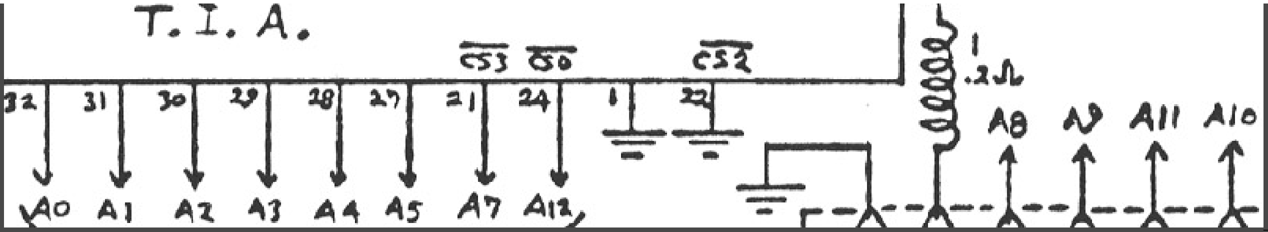
2 - In a NAND chip (74LS00), one of the gates is tied to ground and combined with the signal processor Clock 1 out and the output ground the RDY pin, what makes the processor enters in HALT mode.

Why we need the Clock 1 out?
According to the datasheet of 6502, the processor can not be stopped in a "write cycle". This probably refers to some processor's instructions. Continuing to quote from the datasheet, the state of the RDY should be changed when the Clock 1 Out is at low level. As I understand, it only lowers the Clock 1 out when is out of the "write cycle", indicating the exact time that the processor can be stopped without affect the software running.

Well, the inevitable question: Did work in a regular Atari?
Not exactly, because the 6507 Atari's processor among other things, doesn't have the Clock 1 out. So would be impossible to determine the right time to pause, as the Onyx does.

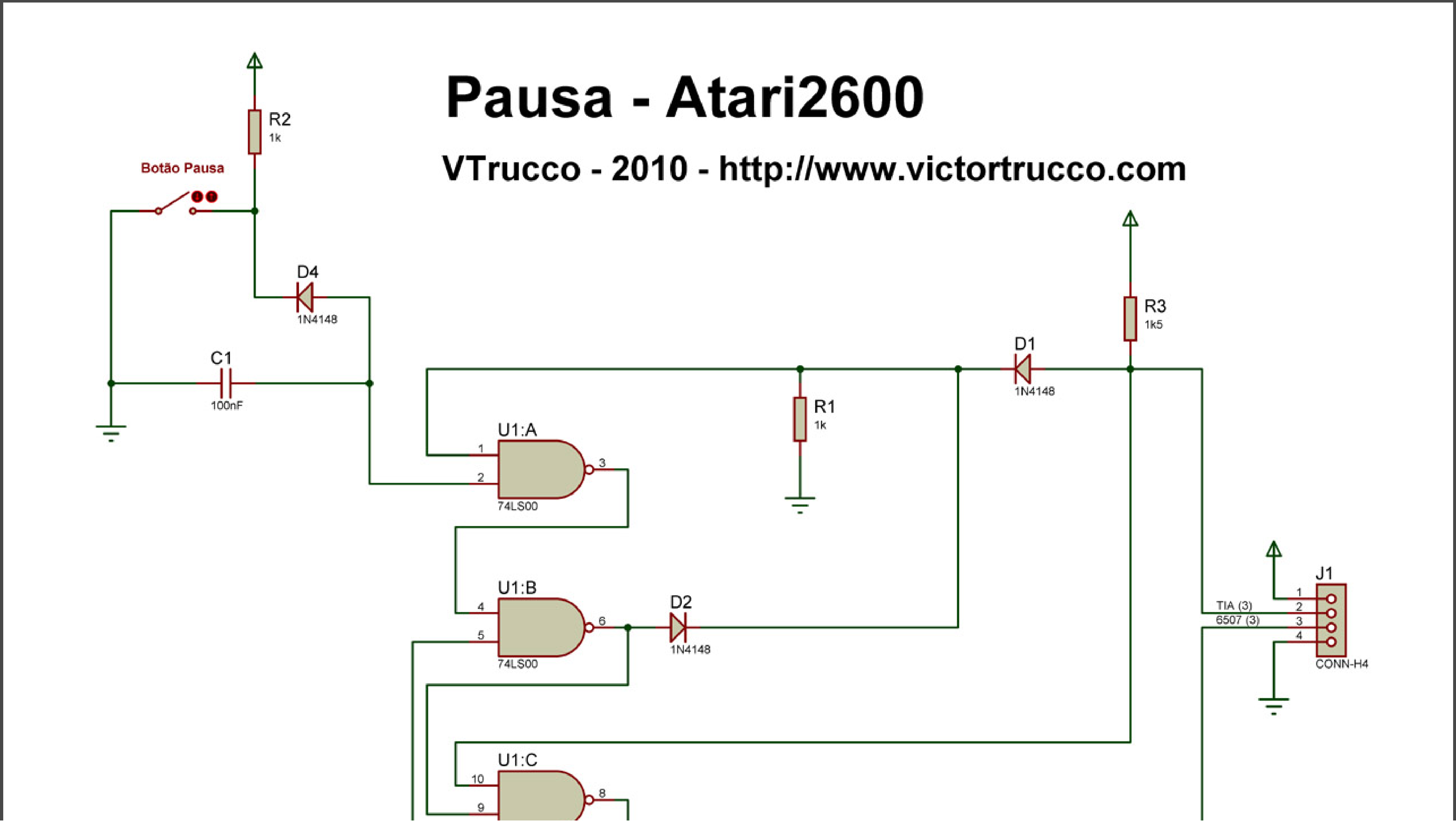
Later, looking at the Atari documentation, I saw that the pin 3 (RDY) CPU was connected directly to pin 3 of the same name in the TIA.

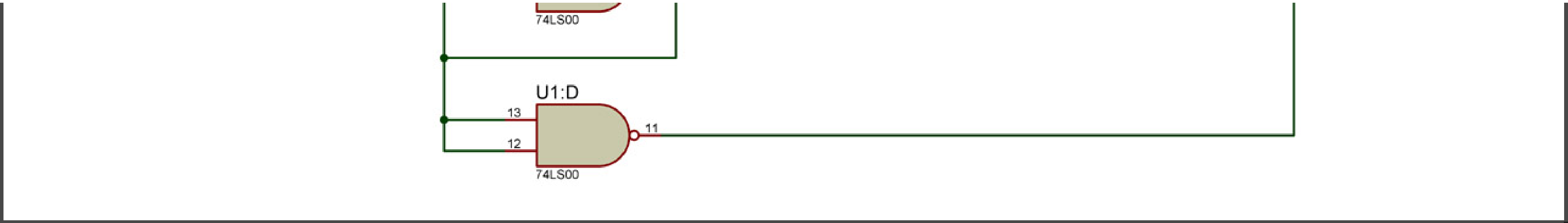




My first idea was something like a reset, but with a logic probe I saw a pulsing signal, so, TIA stops the processor from time to time (several times per second). Many people should already knew, but I with my little knowledge about Atari still didn't know until that moment. Well, the question was ... TIA knows when the CPU can be paused, then why not match our pause signal with the TIA's pause?

In search for the answer, I mounted a little circuit on a breadboard that when triggered the pause button, before the processor gets it, it waits the TIA RDY signal, ensuring the right time to activate it. For the joy of those who posted the initial message in 2005 or who is waiting 30 years to be able to pause an Atari game, **WORKS 100%!**



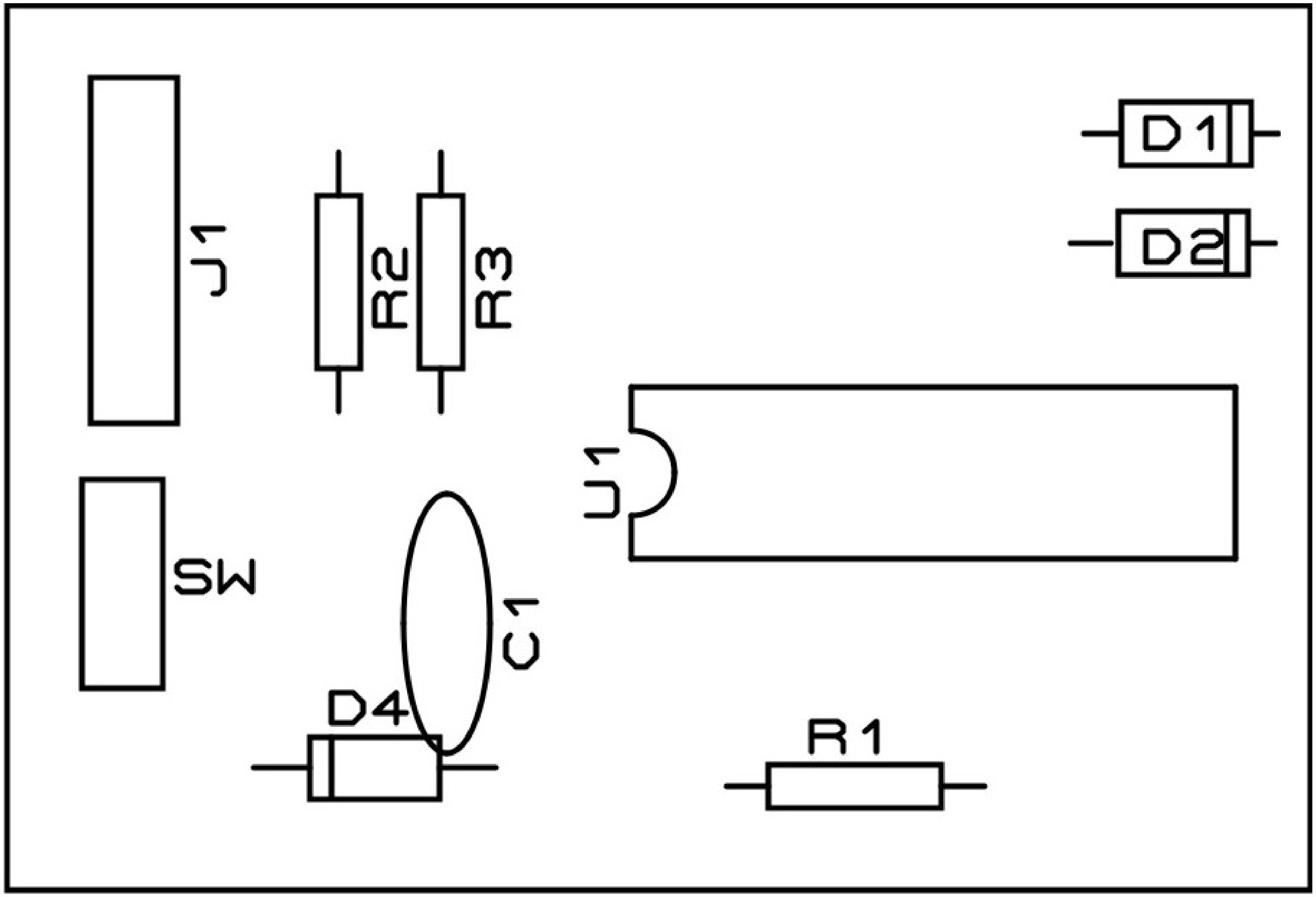


Of course when I say 100%, I haven't tested ALL Atari 2600 games, but all those tested, without exception, have been paused and returns normally. In my tests, I leaved River Raid paused for 3 hours! Others I tested: Enduro (more than 1 hour paused), Combat, Pitfall, Basket, Krull, HERO, and Fishing Derby (more than 15 minutes). These same games I tested short pause periods (like 1 minute or less) and even a lot o pauses per second, by turning the switch quickly. In all cases, the games pauses and came back without any problem. It is not possible to ensure that at all the time, all games, ever, will not crash, especially because the "fragility" of the pause. We must remember that there are high frequencies involved and any "noise" can cause the processor crash, locking the machine, but so far all my operation tests were perfect. Anyway, all my tests were done on a single one console, a Polyvox Atari. I don't believe that is different from most others Ataris and clones, since the components are almost the same.

Well, after the theory, it is time to get to work. All components are easily found at any electronics store. The parts list is:

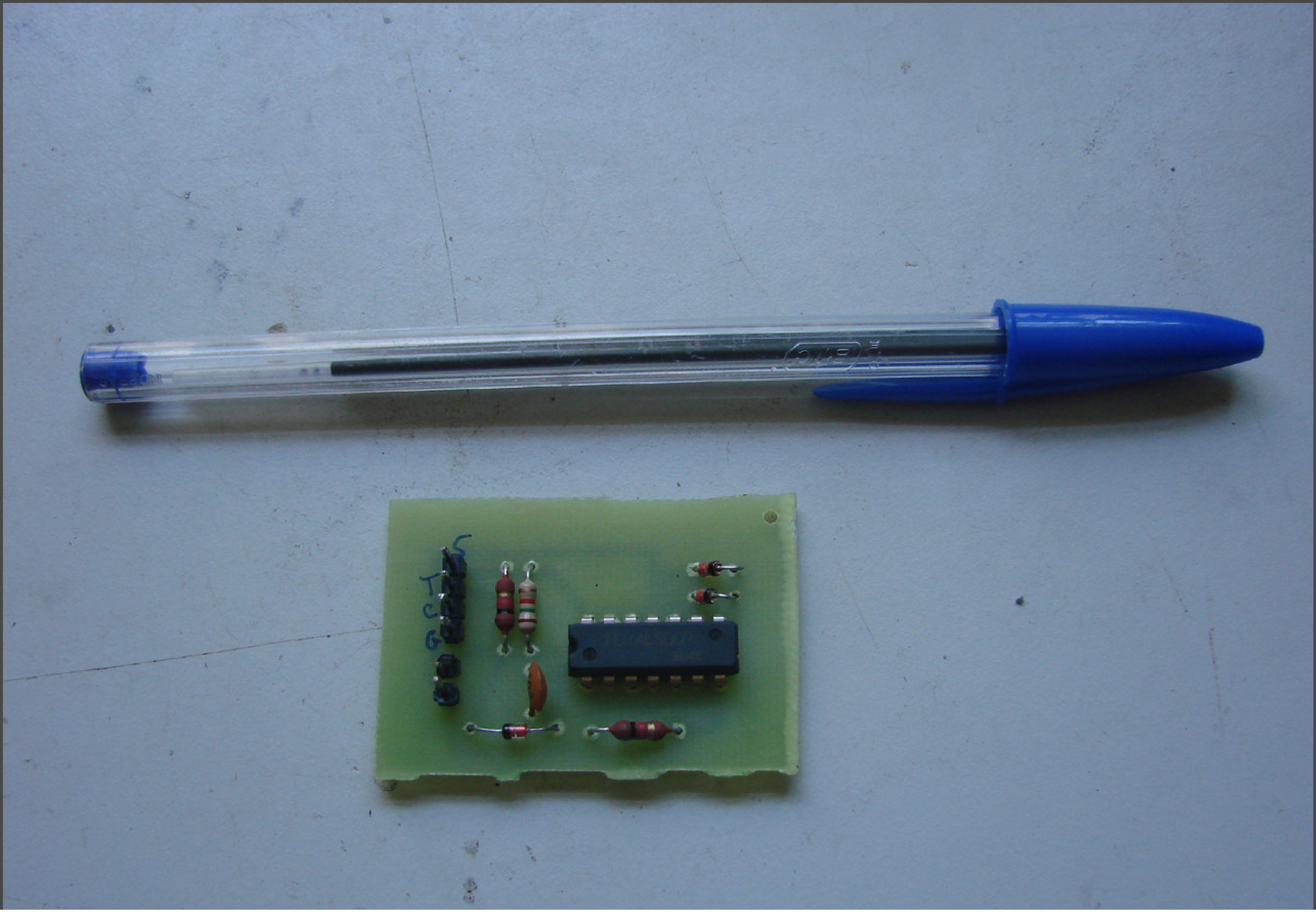
1 74LS00 (I could not make it works with the 74HC00, so buy the LS that is even more "common")
3 diodes 1N4148
2 resistors 1K 1/8W (brown, black, red)
1 resistor 1K5 1/8W (brown, green, red)
100nF ceramic capacitor 1 (usually 104 is written)
1 on-off switch (can be a lever switch or slider)
1 printed circuit board (see below)
Some wire for connections

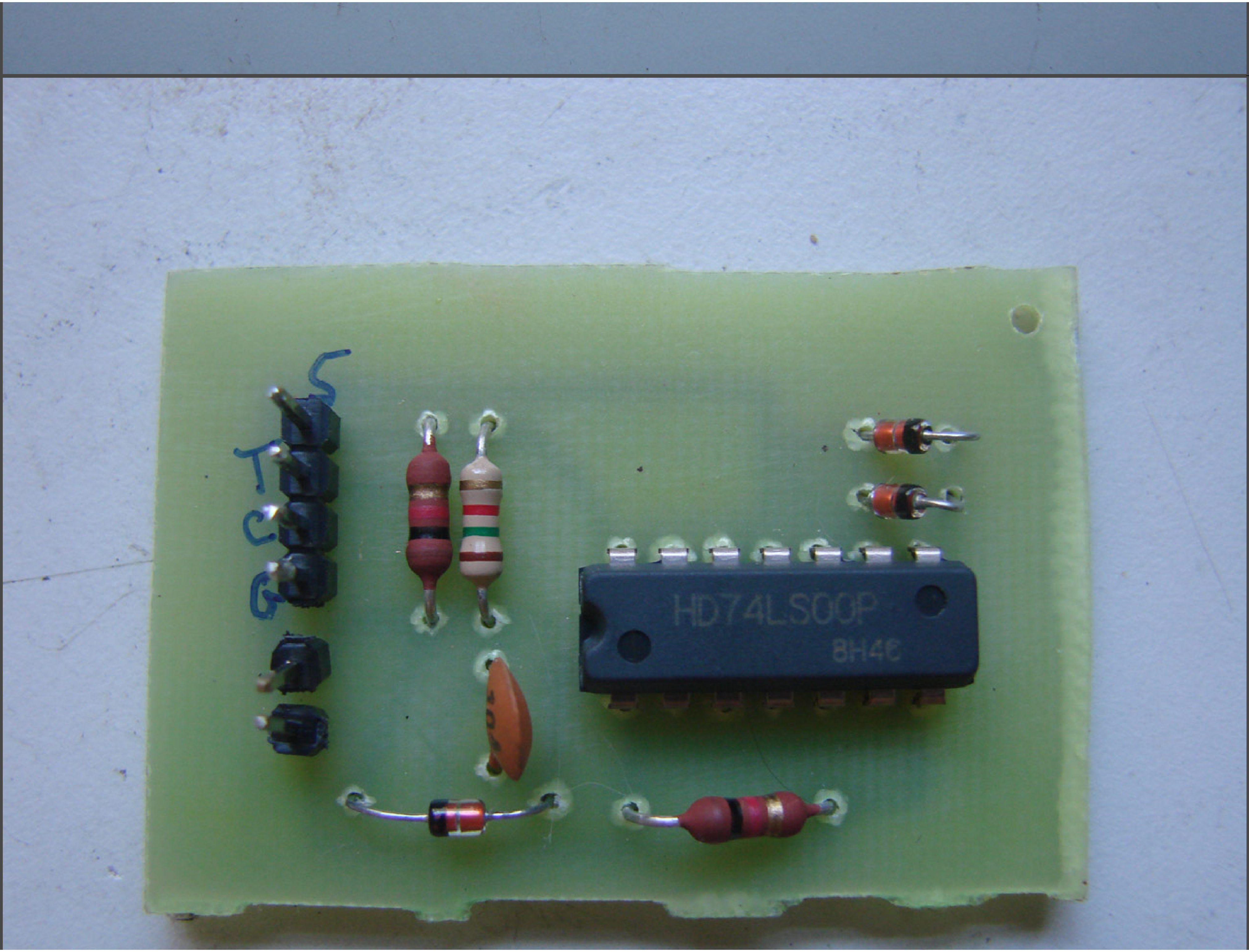
I draw a small PCB and can be downloaded here . It's ready for printing and due to the small size can be easily done by the "toner transfer". For more information, there are a lot of tutorials on the internet. Do a quick search on Google and Youtube that appears much reference material. Please do not blame me for my poor design. This was the second board I draw in my life.

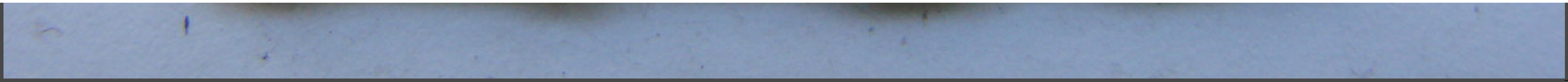

J1
R2
R3
D1
D2
U1
SW
C1
D4
R1

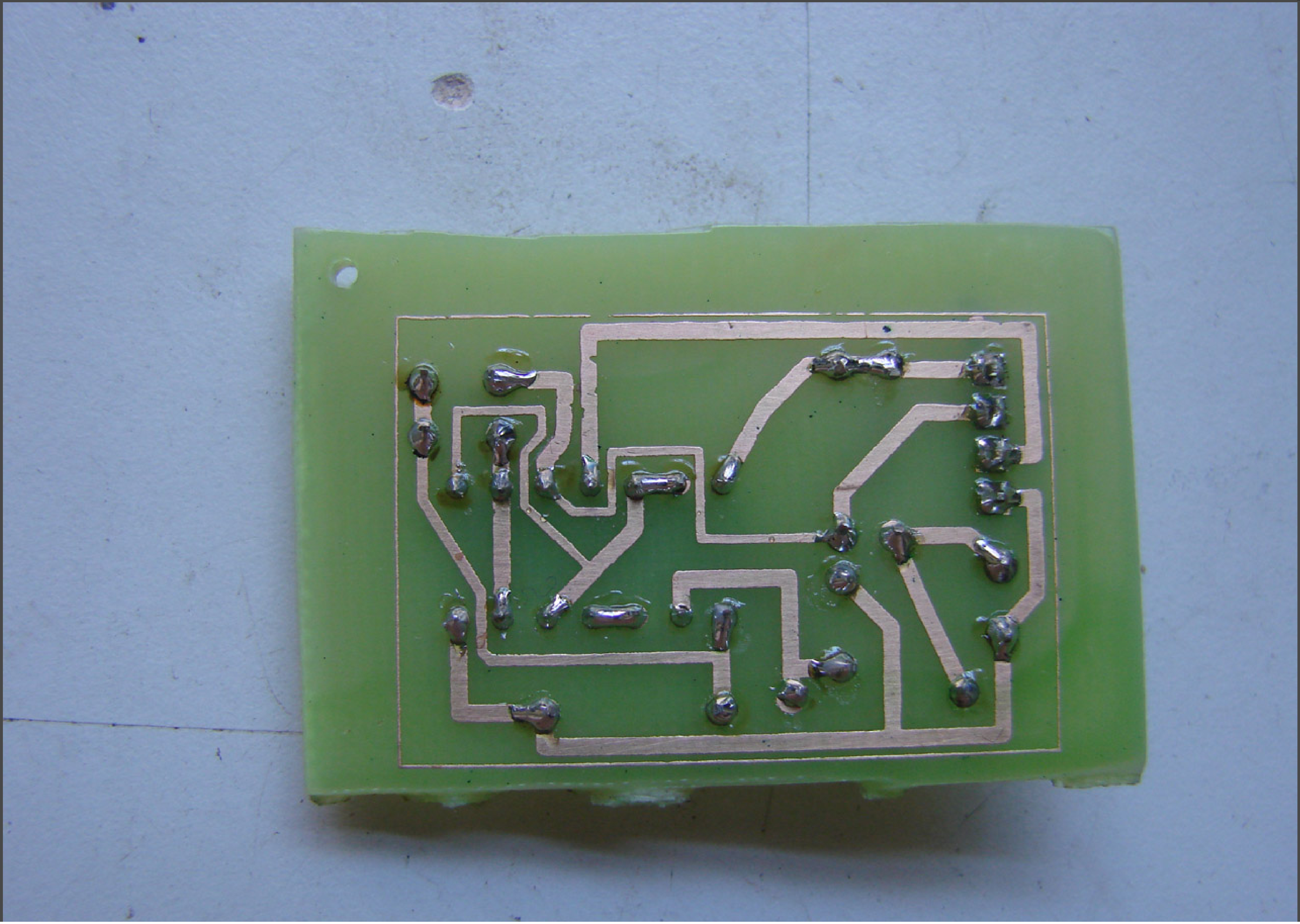
In the picture above we have the arrangement of the parts on the board. Note that the diodes and the IC has the right position to be assembled. The diodes have a ring close one of the termimais and IC has a semi-circle on one side.

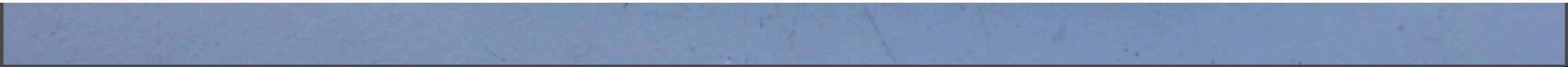
Below, my assembled board. I haven't cut yet because I don't decided how I will mount in the Atari.




S
T
C
G
HD74LS00P
8H46
104

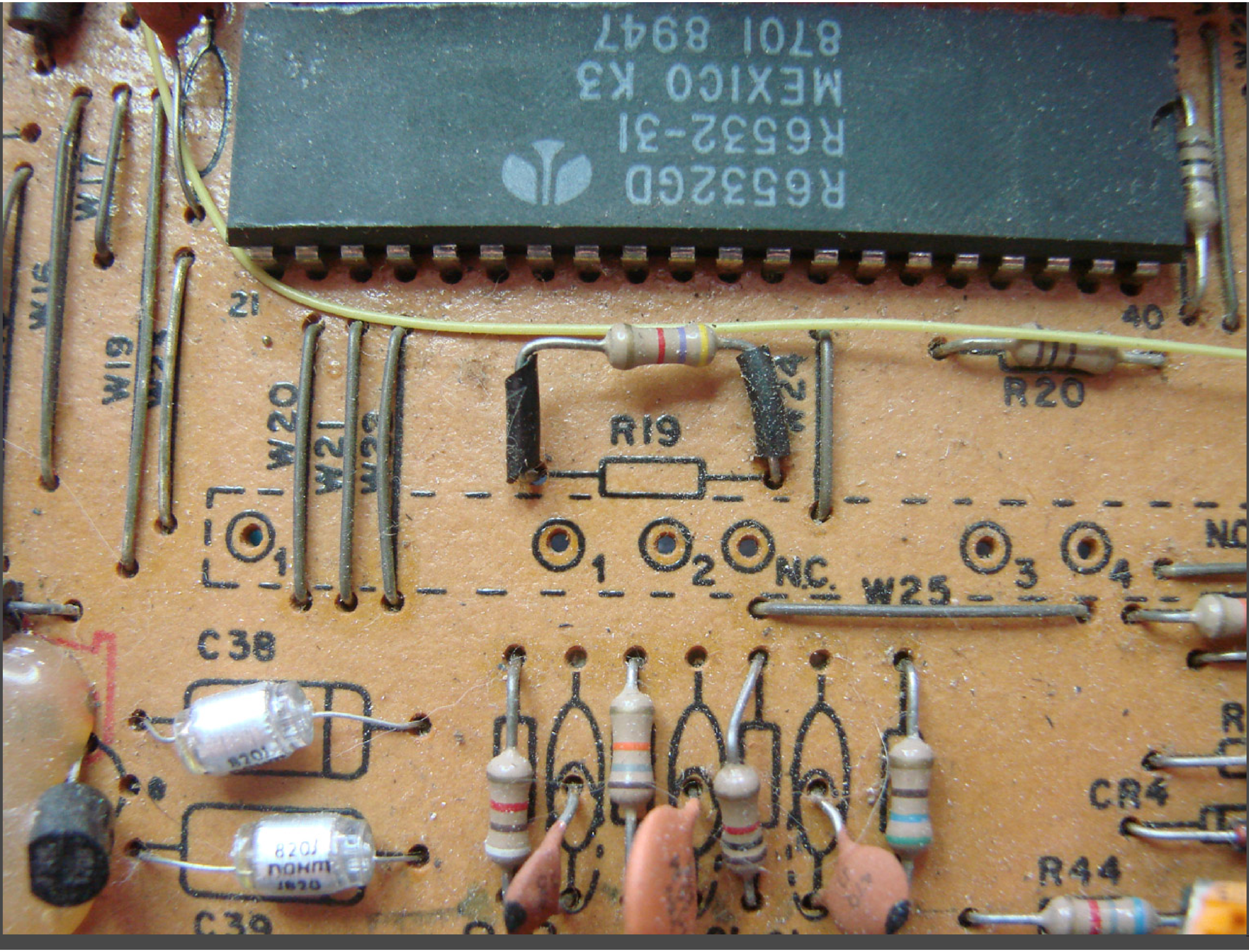


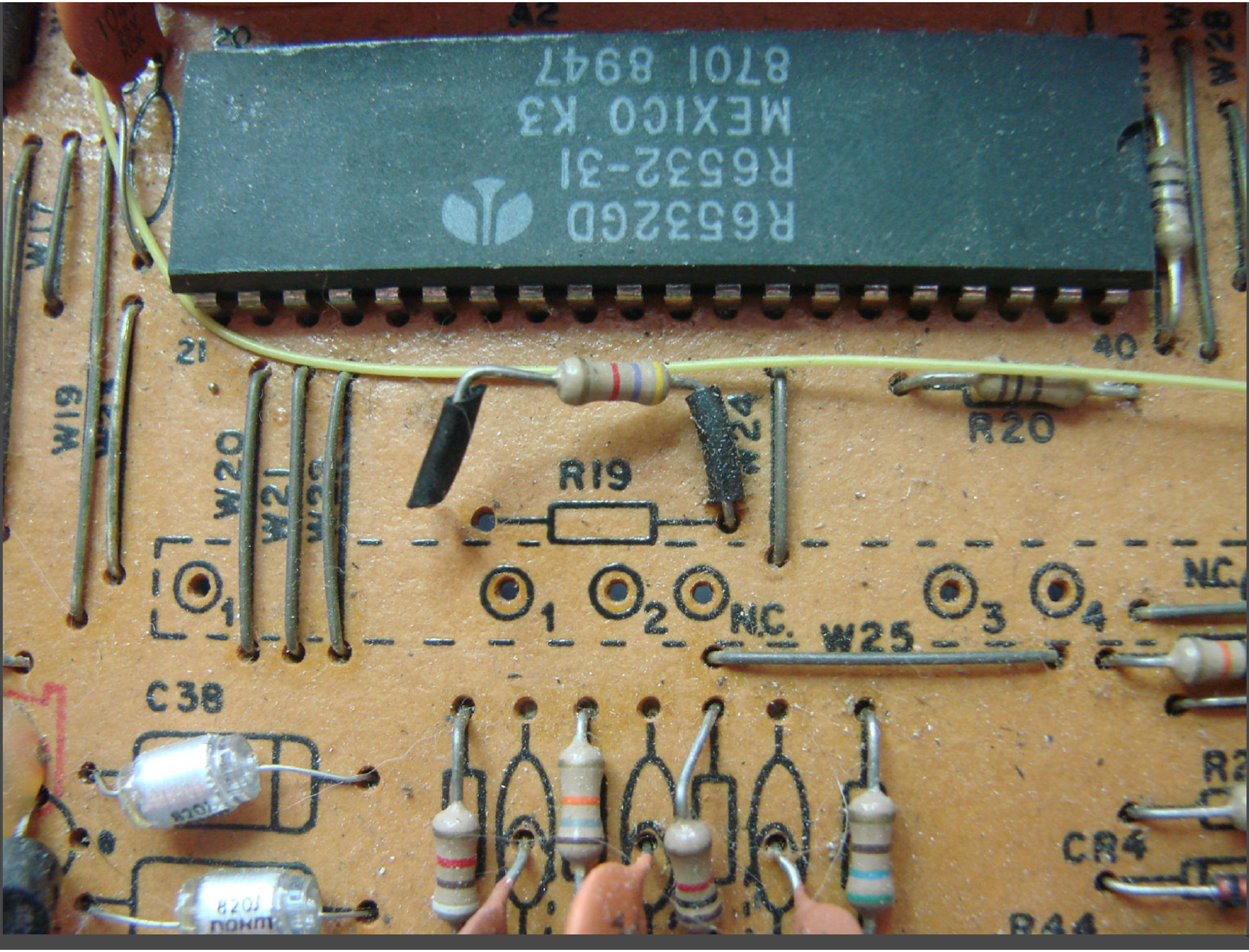




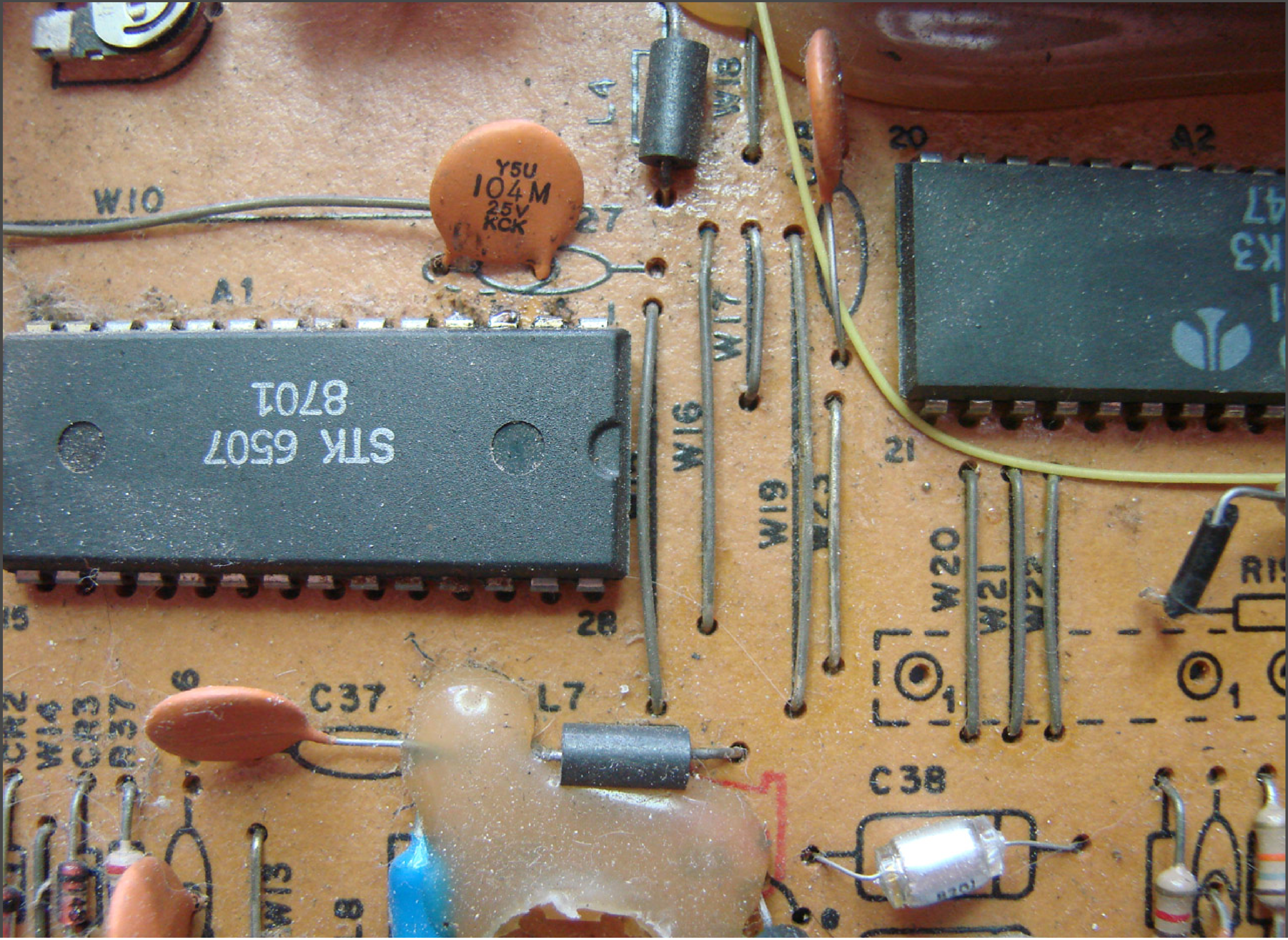
In the space marked "SW", you have to solder two wires going to a switch. This is our pause. You can use a lever switch or a slider switch.

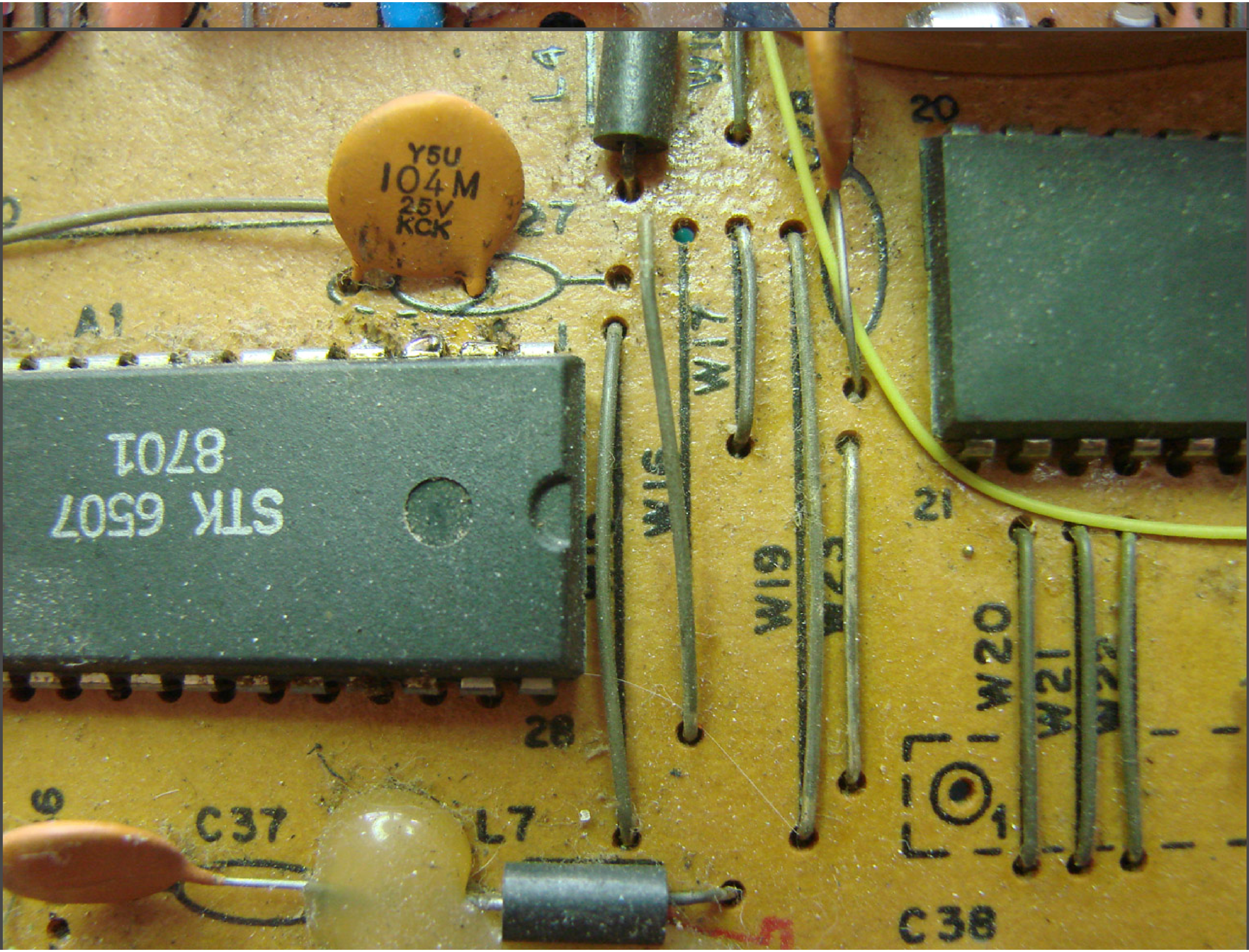
We have to make a small change in Atari. We have to separate the CPU RDY pin from TIA RDY pin and remove the 4K7 resistor that is doing the pull-up in this line.
We will start removing the resistor. In the **Polyvox** Atari it's the R19 in the center of the board. Normally it would be behind the transcoding PAL-M PCB, but I convert this Atari to NTSC. (Complete instructions in my another article. Portuguese only, sorry).


R6532GD
R6532-31
MEXICO K3
8701 8947
21
40
W16
W17
W19
W20
W21
R19
R20
W25
N.C.
1
1
2
3
4
C38
C39
CR4
R44


R6532GD
R6532-31
MEXICO K3
8701 8947
A2
20
1
W28
W17
W19
21
40
W20
W21
W22
R19
W24
R20
1
1
2
N.C.
W25
3
4
N.C.
C38
CR4
R44

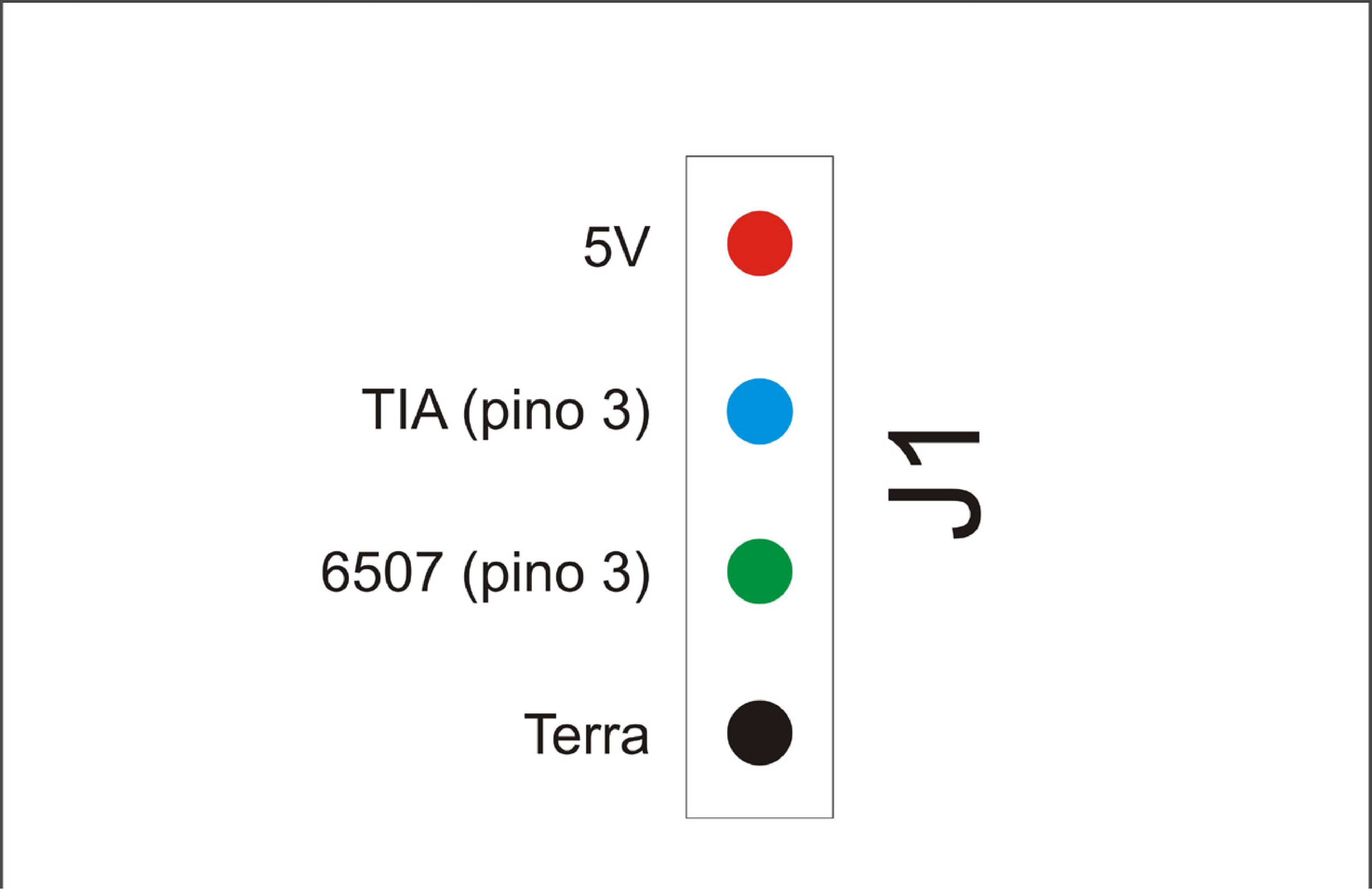
To break the connection between the pins, we will remove the jumper W16 at the right side of 6507.

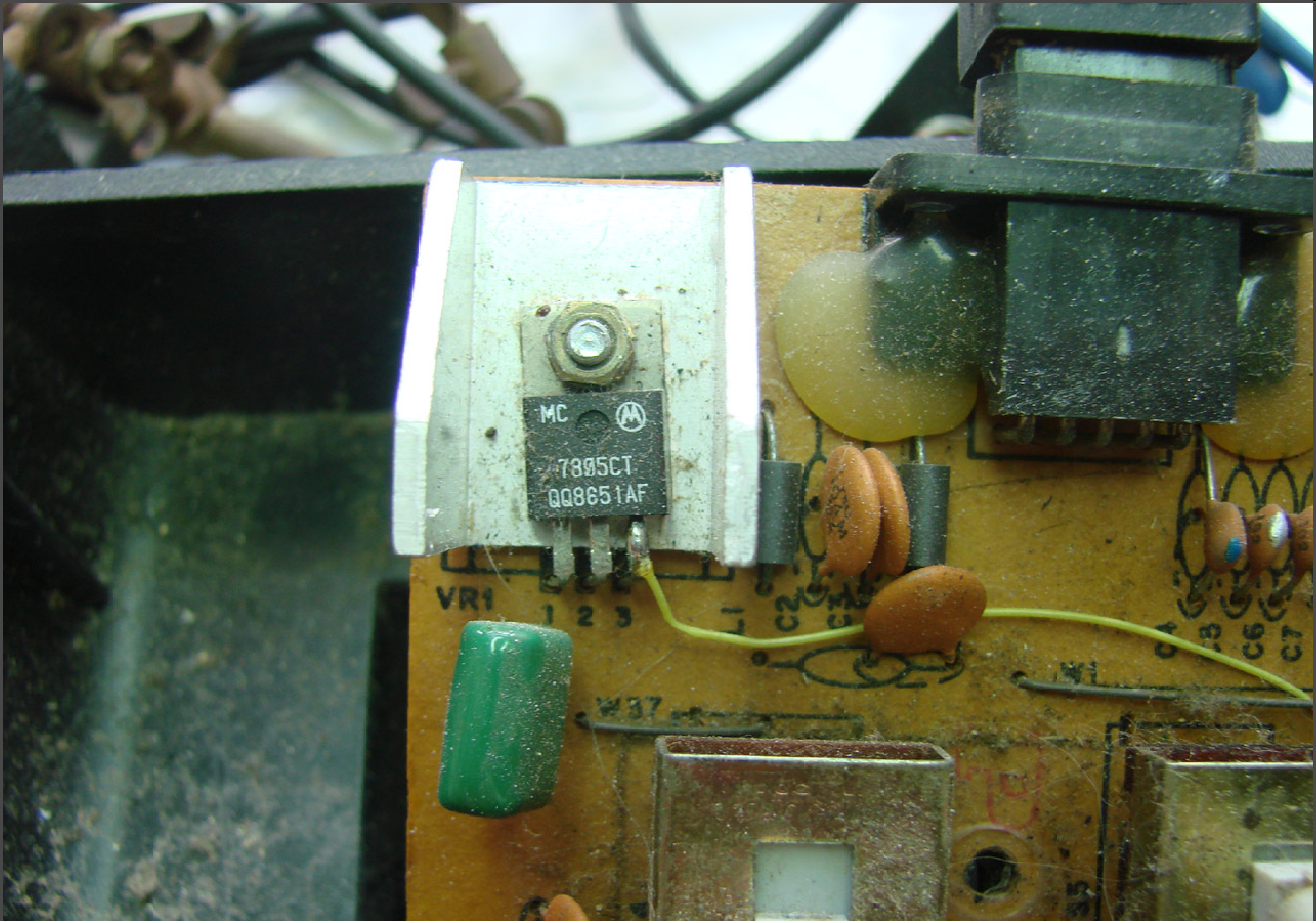


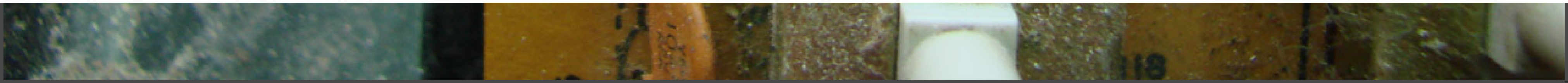

Y5U
104M
25V
KCK
L4
27
A1
20
STK 6507
8701
W15
W17
W19
W20
W21
W22
21
28
C37
L7
C38
1

As I said before, the pictures above are from a Polyvox Atari. If you are doing this mod on another Atari or clone, follow the PCB track that goes from the CPU pin 3 to TIA pin 3. In the middle there is the 4k7 resistor, you must remove it, after that cut the track at any point with a sharp knife. Measure the track continuity with a multimeter after the cut.

Now let's attach our pause PCB. See the image below, the J1 connector. We have to connect the 5V, Ground and the two RDY signals. You need to solder 4 wires in these spaces to connect to the Atari's board. The 5V and the Ground can be soldered directly in the IC 7805, the regulator mounted on a heatsink. The Ground is the middle pin and the 5V is the right pin. In the picture, we can see an yellow wire already soldered to 5V, for the composite video output. (See complete instructions in my other article. Portuguese only, sorry).




MC
7805CT
QQ8651AF
VR1
1 2 3
C2
C4
C3
C6
C7
W1
W37



Be careful to not reverse the TIA wire with the 6507 wire. If linked reversed the Atari doesn't work.

CAUTION: Don't use long wires. They should be as short as possible to not generate "noise" in control signals.

The operation is very easy. In the middle of a game, just turn on your switch to pause the game. Note that the image will get some colorful patterns, but when you turn off the pause, you can play normally again.

If you find any game that does not pause at all, let me know so I can study the causes.