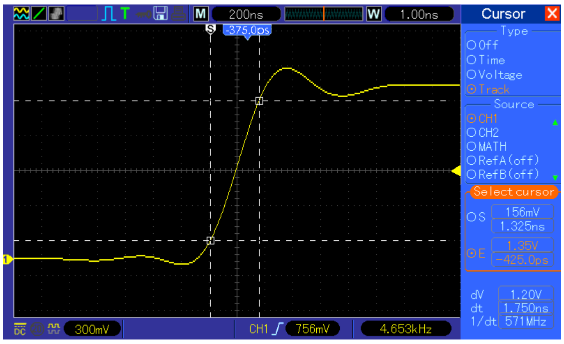
Hantek DSO5062B 250MHz input stage mod

tatus1969, Rev.1, 2016-07-08

It started with the desperate need of a digital storage scope. Wait - I'm 46 now, how did I manage to have lived up to this point with just a 2ch 10MHz analog cathode ray scope? … It was time but not money, so I decided to go cheap and see what I can do to make this one better.

After applying the 200MHz mod to the firmware of this scope (http://www.eevblog.com/forum/testgear/hantek-tekway-dso-hack-get-200mhz-bw-for-free/), I discovered that the analog stage would be far from perfect. As I want to use the scope to be able to observe slope and overshoot of signals from circuits like switch-mode power supplies or motor drivers, I need a scope that does not "add" anything to my signal - except bandwith limitation.

However, after feeding in my first digital signal, it became clear to me that I should not stop at renaming some files...

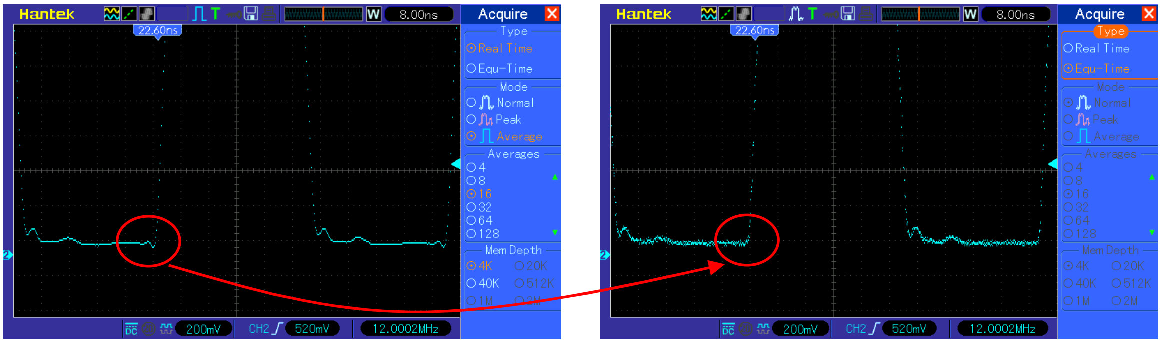


(ehm, sorry, this is not from my scope but from eevblog.org. Didn't take a "before" picture…)

Second (main?) reason for continuing was, could I make even better, and what can I learn here?

While playing around with the scope, I quickly realized that it applies digital post-filtering to the signal before that appears on the screen. The chosen filter (sin{x} / x) naturally introduces visible *pre*-ringing that doesn't come from my signal source. Okay, I thought let's put it into dot display mode, then it will stop doing that. Well, it does *not*.

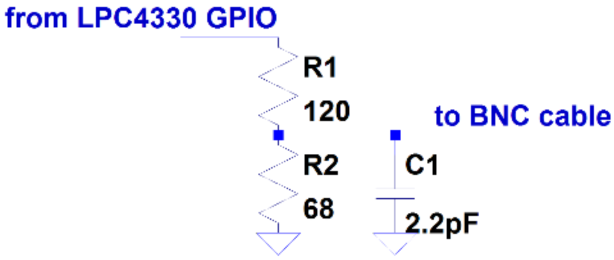
Luckily, they have also implemented "equivalent time" sampling mode for repetitive signals, which samples the signal many times and overlays all the results. This results in a cloud of samples with irregular time spacing between them. But the most important thing is: they finally do not post-filter this cloud but just show it on the screen. I found that additionally choosing "dot" display mode looks best here. The left picture below has been taken with “real time” sampling mode, you see ringing before and after the transitions. The right picture was taken from the exact same signal, but using equivalent-time and dot modes. Compare and decide :-)



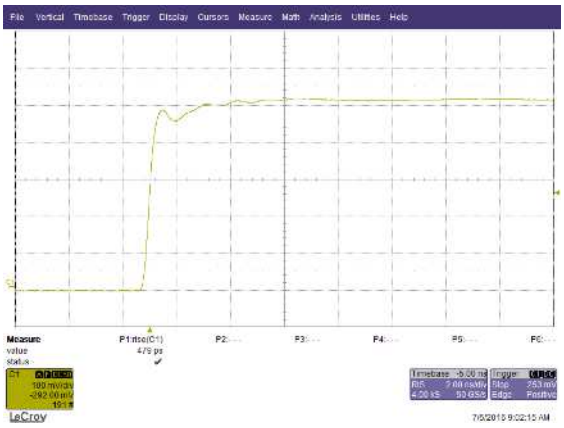
Knowing this and choosing “equ-time” mode from now on, I can be sure that when I see a signal with ringing, it would be either my signal source, or the analog part of the scope.

The next task was to create a pulse signal generator, which would deliver me a slope that would be as fast and ideal as possible. Fast means a lot less than the scope’s original rise time of ~1.8ns, and ideal means no overshoot as far as possible, and as flat as possible after the transition. I thought of building an avalanche pulser like http://www.siliconvalleygarage.com/projects/picosecond-pulser.html, but decided against it, since they seem to be less accurate when it comes to overshoot and flatness. They’re basically ridiculously fast. Well, and I don’t *have* one.

When browsing my “stuff I bought for some project and then forgot about” places, I found an NGX LPC4330 Xplorer board. The datasheet of that micro says that it has some high-speed GPIOs that are specified with 500ps rise time with an output impedance supposedly around 50 ohms. Great! Next, added the following to the output (K6 / SD_SCK):



A measurement with {LeCroy 104Xs with a 1.5GHz + 0.9pF active probe} showed this result (left - direct 50 ohm feed; right - taken with active probe, of course still with terminated cable end):

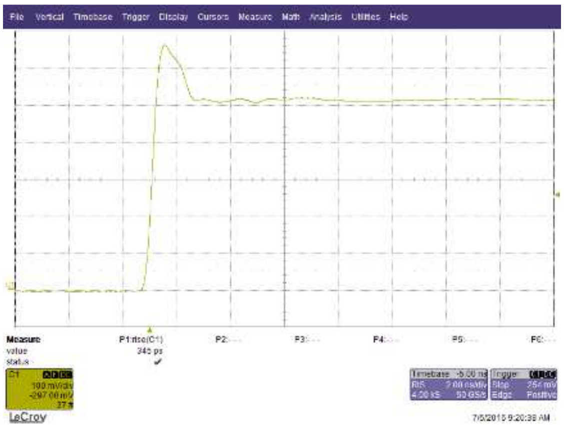

File Vertical Timebase Trigger Display Cursors Measure Math Analysis Utilities Help
LeCroy

Measure P1:rise(C1) P2:--- P3:--- P4:--- P5:--- P6:---
value 479 ps
status ✓

C1 100 mV/div -292.00 mV 191 #

Timebase -5.00 ns RIS 2.00 ns/div 4.20 kS 50 GS/s
Trigger Stop 253 mV Edge Positive

7/5/2015 9:02:15 AM


File Vertical Timebase Trigger Display Cursors Measure Math Analysis Utilities Help
LeCroy

Measure P1:rise(C1) P2:--- P3:--- P4:--- P5:--- P6:---
value 345 ps
status ✓

C1 100 mV/div -297.00 mV 37 #

Timebase -5.00 ns RIS 2.00 ns/div 4.20 kS 50 GS/s
Trigger Stop 254 mV Edge Positive

7/5/2015 9:20:38 AM

So, with light capacitive load (active probe) it apparently does a hefty 345ps. It can almost compete with the picosecond pulser (~200ps), and this is just a microcontroller's GPIO pin. Thanks for that, NXP :-)

But it looks as if the termination on scope side isn't correct? Am I wrong with my assumption of the GPIO port output impedance (the datasheet states a high-level output voltage of VDD-0.4V at 8mA, which would just be equivalent to 50 ohms)? Or is that happening between the GPIO pin and my cable feed network (there's about 20mm of PCB track inbetween)? Maybe it is just the behavior of that GPIO itself. Anyway, I decided to use 50 ohm feed because that ringing looks better and 479ps rise time is still quite nice.

Now I could start digging into the input stage, up to the point where the signal is fed into the ADCs. The reverse-engineered schematic http://elinux.org/images/c/c6/Das_oszi_schematic.pdf helped a lot here.

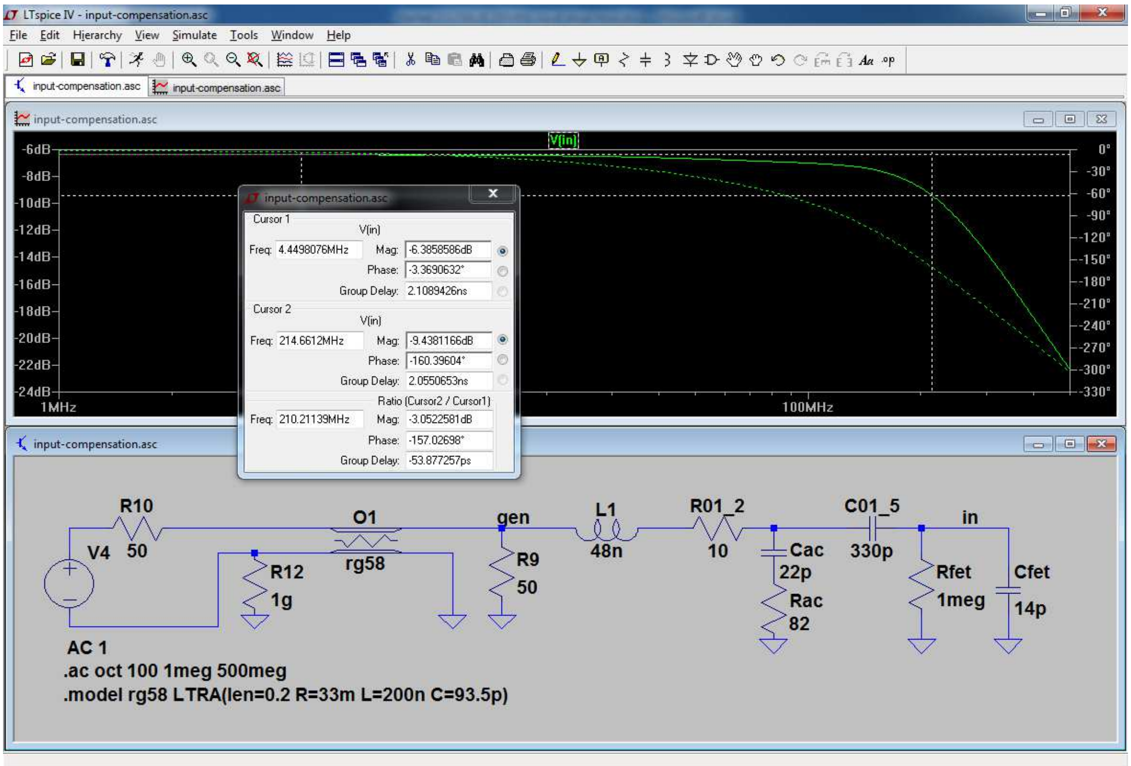
My scope came shipped with the following component values (I only list the designators for channel A here and in the following for simplicity):

R01_2	= 30R
Q01_1	= MMBF4392
Q01_2	= MMBTH10LT1
RX1_1..4	= 280R
RA01_1/2	= 22R
U01_3	= LMH6522

I did some measurements and simulations, and after some experimenting I came to the following set of modifications. In some cases they contradict tinhead's approach in http://www.mikrocontroller.net/attachment/173049/mod_input_circuit.pdf, and in each case I'll explain why.

1. R01_2
   The first part of the input stage – from the BNC connector to the JFET – actually took me the most time to solve. The situation at the beginning was that, even though my test signal would be perfectly fed in at 50 ohms without distortion - when it arrived at the JFET it already had significant ringing. The problem with this part is that the signal runs through quite long PCB traces (creating inductance), including two (non-RF rated) mechanical relay contacts. They could have done a lot better here, but I also didn't want to sacrifice functionality like 1:10 attenuation by just creating a minimal-length bypass. With the help of a minivan I could measure 14pF

capacitance (final circuit with MMBJF309) and 48nH inductance. This value is surprisingly high, I guess the relay accounts for a great deal of that. I used LTSpice then to find the correct compensation with maximally flat response and at the same time minimal bandwidth reduction. The result is shown here (R01_2, Cac, Rac):



input-compensation.asc

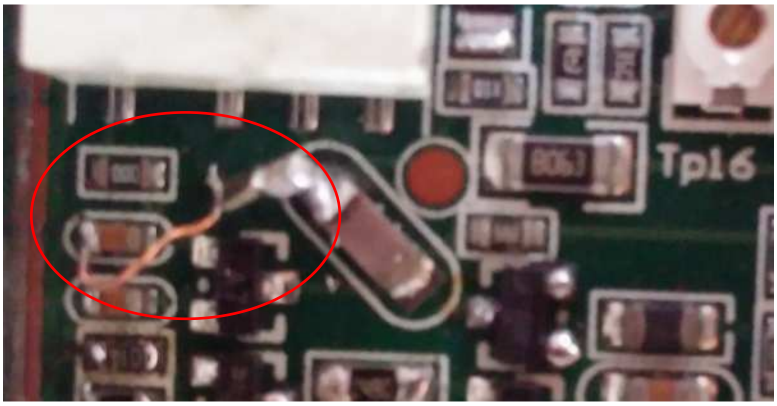
Cursor 1
V(in)
Freq: 4.4498076MHz Mag: -6.3858586dB
Phase: -3.3690632°
Group Delay: 2.1089426ns

Cursor 2
V(in)
Freq: 214.6612MHz Mag: -9.4381166dB
Phase: -160.39604°
Group Delay: 2.0550653ns

Ratio (Cursor2 / Cursor1)
Freq: 210.21139MHz Mag: -3.0522581dB
Phase: -157.02698°
Group Delay: -53.877257ps

```
AC 1
.ac oct 100 1meg 500meg
.model rg58 LTRA(len=0.2 R=33m L=200n C=93.5p)
```

Besides changing to **R01_2=10R**, I added a **Cac=22pF** capacitor in series with a **Rac=82R** resistor. The one end goes to the C01_5 (signal “Tp101”), the other to the closest available GND. Here’s a picture of that mod:



2. Q01_1
Not too bad, but this is not a dedicated low-capacitance RF JFET. Replaced with **Q01_1=MMBFJ309** (Ciss ~ 8pF). Thanks to Fixup for that tip.

3. Q01_3, Q01_4
These ones form current sources for the amplifiers. The better the responses of them, the more constant the currents are, and the more linear the amplifiers will be. BC846B’s are very cheap, but simply not RF. Replaced both by **Q01_3=Q01_4=MMBTH10LT1**.
P.S. I also tried a MMBFJ309 based current source. That was a bit faster, but didn’t justify the additional effort - that JFET cannot directly go on the BC846B land’s, it requires some nasty “free-air” “single-leg” soldering magic.

4. R01_25
   I completely fail on this one. Does anybody know why they could have introduced this resistor? Is that an attempt for Vbe temperature compensation (what about hfe variation then)? Did they want to couple both current sources to improve transient behavior? Changed to **R01_25=0R**.

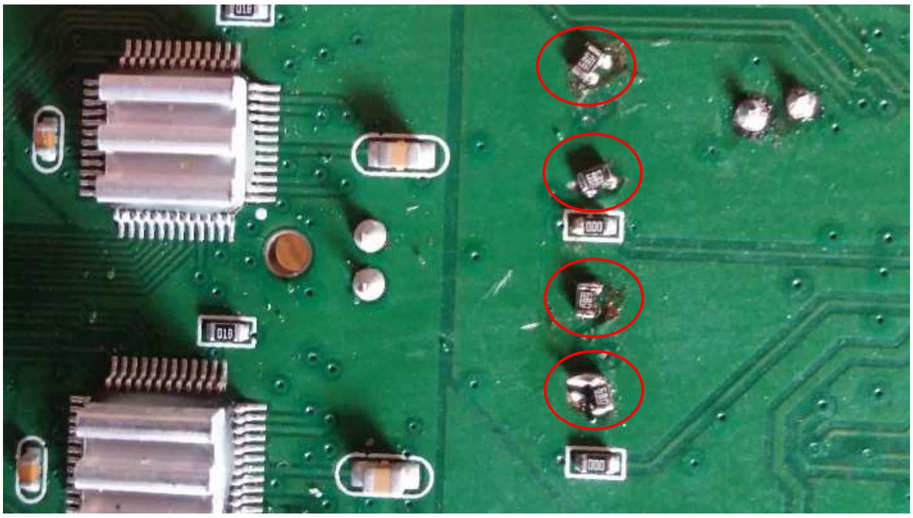
5. R01_26
   The new MMBFJ309 JFET has a zero-gate drain current of >=12mA. We must not exceed that value, or the gate will eventually become forward biased – and conductive. For 10.8mA nominal current, **R01_26=120R**.
   By the way, R01_27 / R01_29 are fine as they are in my opinion. Of course they limit the amplifier's dynamic range, but they also help the current sources maintain their set points. In no way they affect the input stage's frequency response.

6. R01_27 / R01_29
   Gave it a try and changed to **R01_27=R01_29=0R**. But can someone help me understanding why this is increasing bandwidth (not much, but noticeable)? With the 250R's still in, the only difference should be a different DC operating point for the current source transistors. The currents should be even more stable…

7. D01_2
   After my first tests and measurements, I removed it out of suspicion. I will put that back in to see if it limits BW where it should not, but properly driven it should not. Listed here for completeness: **D01_2=dnp**.

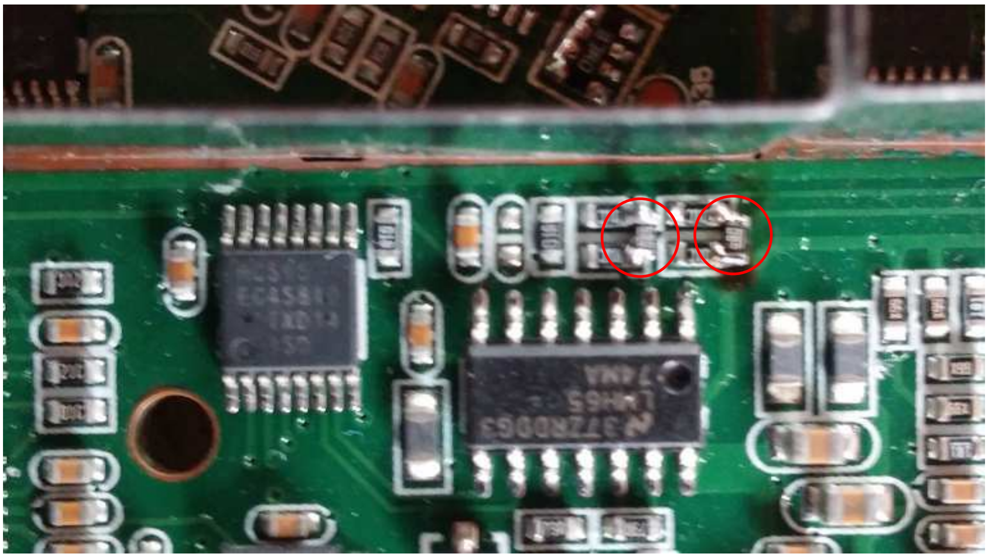
8. RX1_1..4
   When looking into the LMH6552's datasheet, I noticed that that amp would produce quite some ringing depending on the chosen feedback network (page 10, fig.9). The amp has 1.5GHz bandwidth, so there should still be plenty of headroom even when made slower by increasing these resistor values. For the following step I also needed to raise its gain from 1 to 2, so **RX1_3=RX1_4=390R**, and **RX1_1=RX1_2=180R**. Should be 0.1% tolerated parts according to LMH6552 datasheet.

9. RA01_1/2
   On its way from the LMH6552 to the ADCs, the signal runs through differential transmission lines. But as they are not terminated at the receivers, we can expect reflections going back. This is worst for channel B, because the PCB tracks are longer here.
   Even if RA01_1/2 had been chosen to match the transmission line's wave impedance (which they weren't) and could perfectly dissipate the reflected wave, the LMH6552 would still need to provide the current pulses required for that. Yes, it is fast, but it cannot be as fast as a *resistor*.
   I chose to try it the "right" way and actually terminate all ends. I realized that I cannot actually terminate right at the ADC's inputs, because the scope parallels all four ADCs when it needs 1Gsps for a single channel, and this would also parallel my terminating resistors. The closest possible location is underneath the two relays doing that multiplexing (RL04_1/2). I had to carefully scratch off solder resist from the corresponding vias at the board's underside, and solder the resistors right across them. The result looks like this (input stage is to the right):



Finding the correct value that matches the line's wave impedance was done using potentiometers. It came out with something around 65 to 70 ohms. So my choice was **Rterm1..4=68R**, and **RA01_1/2=33R**. Of course this also introduces 1:2 attenuation, which is why I had to raise the LMH6552's gain in the previous step. All parts should again be 0.1% if possible, as that helps in keeping their attenuation closely matched. In 1Gsps mode we need to make sure that all ADCs see exactly the same signal.

As the trigger stage also receives a copy of the signal, we need to terminate them with **Rterm5/6=68R** (0.1%) as well. This mod is fortunately much easier as shown here:

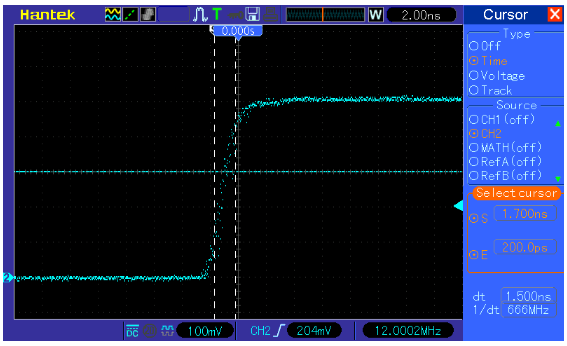


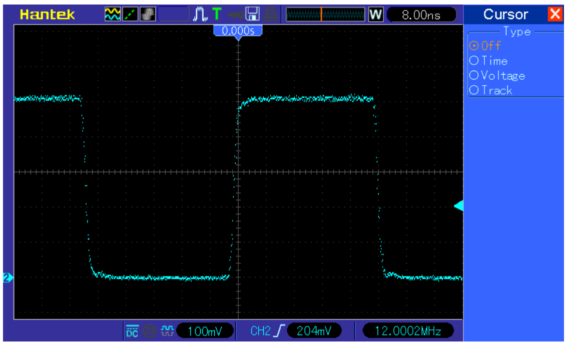
Remaining work:
- need to adjust compensation of internal 1:10 input attenuator

The reward of all this hassle is a USD300 budget scope with **1.5ns** rise time and **250MHz** analog bandwidth, and - for me even more important - with close-to-**Gaussian** response: seeing overshoot only if that is *in the signal*.

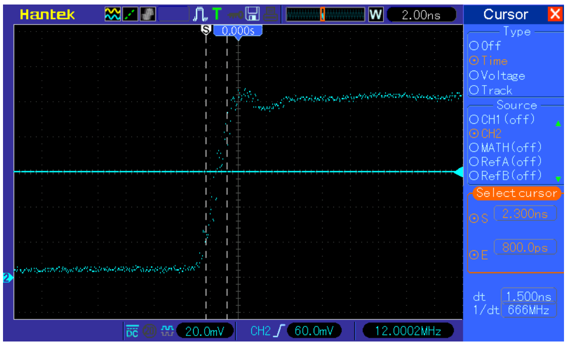
I hope you enjoy the following series of pictures, taken with this device and my 500ps pulse generator.

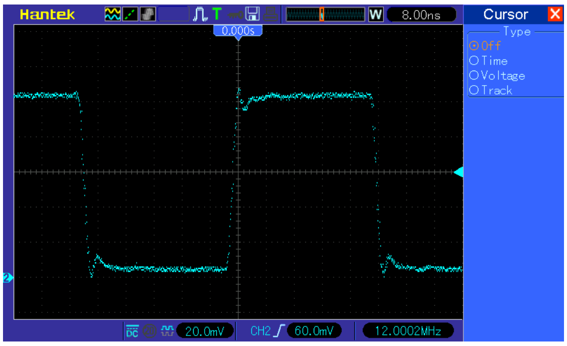
Direct connection with termination at scope input (BNC-t + 50 ohm stub):





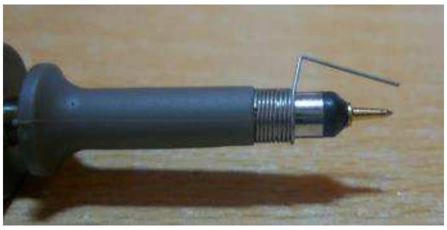
Tektronix P6205 active probe (750MHz / 2pF) (used @ ebay ~ 50USD), 50 ohm termination at scope input (BNC-t + 50 ohm stub):



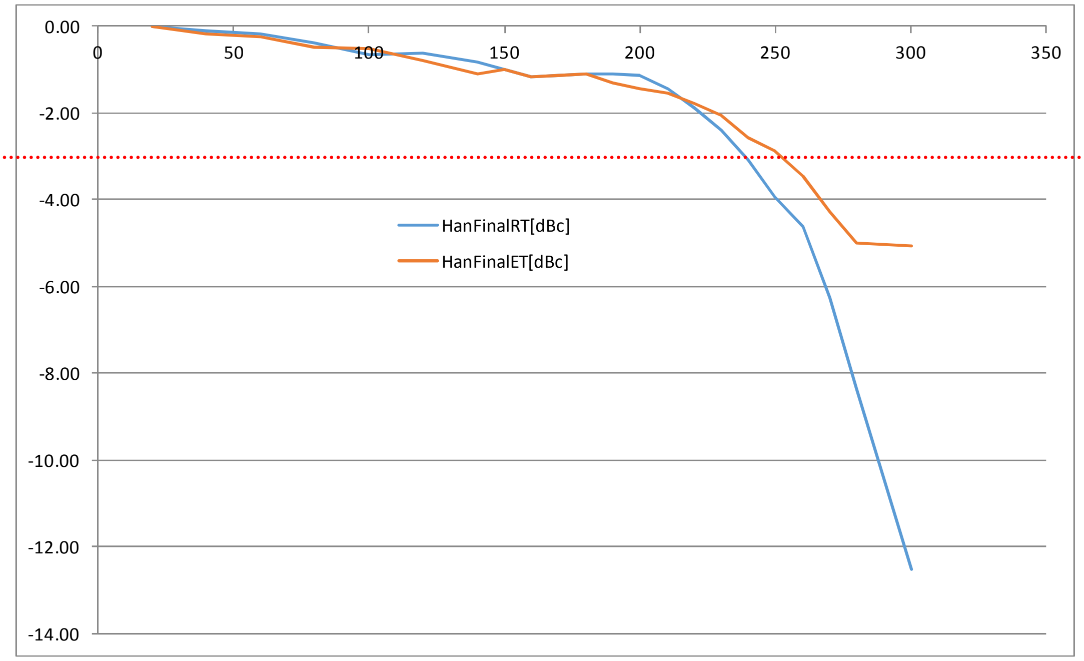


Not 100% flat, but the test signal isn’t either. It actually looks pretty similar to what the LeCroy sees.

By the way, I did pay attention to use low-impedance probe grounding, similar to what is shown here. Otherwise the result would have been waaaay worse.



Last but not least, I used the miniVNA to make a frequency response measurement. The problem with that device is that its output stage is not very stable in amplitude, nor in impedance. But I think it is enough to judge the achieved bandwith of this mod; I made reference measurements with the LeCroy to at least compensate for the amplitude error. The next picture shows two measurements from the Hantek: one with realtime acquisition mode ("RT"), the other with equivalent time ("ET"). The bandwidth limitation effect caused by the post-filter can easily be seen.


HanFinalRT[dBc]
HanFinalET[dBc]