EEVblog Electronics Community Forum

A Free & Open Forum For Electronics Enthusiasts & Professionals

Hello horo

Show unread posts since last visit.
Show new replies to your posts.
March 05, 2025, 08:22:14 am

Entire forum Search

News:
Be excellent to each other.

Home Help Search Profile About us My Messages Calendar Links
Members Logout

EEVblog Electronics Community Forum » Profile of rpcope1 » Show Posts » Messages

Profile Info Actions

Show Posts

This section allows you to view all posts made by this member. Note that you can only see posts made in areas you currently have access to.

Messages Topics Attachments Thanked Posts Posts Thanked By User

Messages - rpcope1

Pages: [**1**]

1 **Test Equipment / Re: Hantek 6022BE 20MHz USB DSO**
« **on:** July 28, 2015, 11:33:57 pm »

Quote from: daybyter on May 30, 2015, 05:59:20 pm

> @rpcope1:
>
> Hi!
>
> I managed to get your example going! Great work! I had to install a few packages, though:
>
> That was python-matplotlib , python-scipy, python-numpy, python-pip, python-usb, libusb1 (with pip), python-dev, pypy-dev (not in that order, though...)
>
> I had to add a path to the usb lib:
>
> PYTHONPATH=$PYTHONPATH:PyHT6022 python examples/example_linux_scopevis.py
>
> And got some error from time to time:
>
> raise ValueError("x and y must have same first dimension")
>
> ```
>  File "examples/example_linux_scopevis.py", line 28, in <module>
>     pylab.plot(timing_data, voltage_data, color='#009900', label='Raw Trace')
> ```
>
> But now I got a wonderful trace!
>
> You want to translate your code to C now? Or just create a driver for sigrok?
>
> Thanks for your great efforts!
>
> So I googled my way through python (I don't really like python) and extended your linux vis example by a few line to animate the waveform. I did not use any python bitblit methods for now. Just clear and redraw the screen. So I got maybe 2 to 3 updates /

> second in my virtualbox now. I added some very simple code for triggering, but it doesn't really work yet.
> Not sure, if it's actually worth to work on this python code for now, but better do some C++ coding (qt maybe).

My apologies, I don't check this often. I am considering translating some of it to C; you could also create a sigrok driver, but my understanding is most of the scopes sigrok supports have built-in triggering, which will be a major challenge here. The enormous limitation on this device is that unless you add an external trigger, triggering needs to happen in software, which is going to mean writing code that has some level of parallelism involved (need one thread to poll USB data, one thread to handle trigger, one thread to handle display). This is problematic on CPython because the GIL limits thread based parallelism, meaning you either need to rewrite the hotspots in C or Cython. I am currently leaning towards Cython, because the code is simpler to maintain, and has most of the C performance gains. I had not considered writing an API for anything other than Python, but maybe in the future the work can be ported to an API in C or C++. Someone with some familiarity with libusb could do most of the work pretty easily.

As far as resolving path dependencies, did you run "sudo python setup.py install" before running the examples? That should install the dependencies into your site-libraries folder, and you shouldn't need to modify the PYTHONPATH.

Reply Quote Notify

2 Test Equipment / Re: how to save Hantek 6022BE scope data to .WAV

« **on:** May 02, 2015, 09:47:43 pm »

Open6022BE was done by RichardK, but I've got my own repo to do that, and if you look at https://github.com/rpcope1/Hantek6022API/blob/master/examples/example_trezor.py or https://github.com/rpcope1/Hantek6022API/blob/master/examples/example_linux_trezor.py (depending on if you're on Windows or Linux), that might help you get where you want to go, if you feel comfortable with Python.

Most of my work will be primarily Linux focused for the time being, but if you have the stock drivers, you should be able to utilize the Windows examples from when I first wrote the repo.

Reply Quote Notify

3 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** April 28, 2015, 05:50:47 am »

> **Quote from: baruch on April 28, 2015, 04:50:00 am**
>
> > **Quote from: rpcope1 on April 28, 2015, 01:55:57 am**
> >
> > So firmware isn't so bad on this device. If you look at my repo, it's got the hex code for the stock firmware; the device has no sizeable non-volatile storage to speak of, so the original Windows driver sends the firmware over USB every time the device is plugged in. I was able to capture the data by running Windows in a VM and running a trace on the USB port in wireshark. The MCU that controls the device in the scope is a CY7C68013A (FX2LP), which is really ubiqitous. It sports a lot of GPIO, an 8051 processor and a 4K FIFO that buffers the data. The ADC is an AD9288, which is also as common as dirt. If you're interested in the firmware, i hope you like assembly 😜. Cypress kind of suggests you use Kiel but I am using mcu8051ide, and there are lots of open source 8051 tools including disassemblers out there; the instruction set on the device is also pretty sane. Pretty much everything use in the scope is well documented, and I can provide further details if you have questions.
>
> After I posed the former message I found your repository, I still don't have the device to really play with it but started to look at the assembly. I'm adept enough at reading assembly but never really tried writing it beyond adapting sdcc peephole rules for optimization.
>
> You are currently patching an existing firmware, I personally would have gone for understanding the pinouts and interface points and writing a new firmware and then you can do the writing in C to reduce programmer effort.
>
> I would also avoid writing a whole new scope software, there are already several for the 6022BE, the RichardK one and others as well. I personally would prefer to just write a driver for sigrok and let another project handle the GUI.

There were two issues with using the RichardK software moving forward: it's built for Windows (all I have is a

Windows XP VM, every PC I have is Linux/BSD, so having something Linux specific is important), and it utilizes the Hantek driver (as I understand) which operates rather differently than what I'm doing (the biggest thing being you'd need to add threads for async, which may or may not be problematic). I will see what it would take to port it, though; what would be more interesting is porting the drivers to sigrok. I need to look at how they do things, and will when I get time. I agree, it's probably better to consolidate efforts on the front-end stuff. As far as patching the existing code, I agree that green field development would be a good way to go, but Jochen has been making those patches, and I won't argue with his awesome work there.

Really the ultimate goal for this repo is to provide a nice Linux/POSIX (and hopefully Windows) API to the device, and any front-end, if any, comes second; I agree though, once I get a better feel for what's possible, I think we can move towards building drivers for things like Sigrok and OpenHantek, so that people can get use out of the device and leverage others well tested front-end code.

Reply Quote Notify

4 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** April 28, 2015, 02:04:26 am »

Quote from: 2010kira2010 on April 27, 2015, 10:53:14 am

> Hello everybody!
>
> Please!!!!!
>
> Who can copy the firmware eeprom?

No need to copy the EEPROM, it's probably the most useless chip on the board. It doesn't hold any firmware, it just "stores" calibration values that the stock driver uses (and I do not). All of the firmware is transferred by the host everytime the device is plugged in and held in memory; you have to flash firmware every time you power the device up. If you look at my Python API, methods are provided to both read and write the EEPROM if that's your thing.

Quote from: momus on April 27, 2015, 05:15:18 pm

> Rpcope, your research on this scope seems -very- promising.
> Your python api seems to be easy to use, I thought I wouldn't have a clue what your examples are doing, but it's quite readable.
> I will maybe try to tinker with it.
>
> Creating a good GUI on top of this could be a huge step forward.
> If you even manage to open the door to a real hardware triggering, I think I'll build a shrine for you...

I had started to hack together a GUI using matplotlib, but I'm fighting with the fact getting parallelism in CPython is not easy; just consistently reading and trigger is probably enough to saturate a CPython process. I am hoping to work to make the documentation better on CPython also; I'd like to get this to the point where someone with basic Python skills can still use this device in a meaningful way. Real hardware triggering might (or might not) be possible, I just need to slog through the 8051 assembly now. It's also possible that we might be able add significantly better software triggering than the stock software (see my above post).

As an aside, honestly this device is probably more palatable if you think about it as a DAQ with few ports, decent speed and marginal precision; to anyone who is reading this if you're thinking about buying this as your primary scope, trust me, save up and buy a Rigol. 🙂

Reply Quote Notify

5 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** April 28, 2015, 01:55:57 am »

Quote from: baruch on April 26, 2015, 11:42:58 am

> **Quote from: rpcope1 on April 24, 2015, 09:15:18 pm**

> > As a more general update, it turns out the firmware Hantek shipped this device with has got some bugs and is otherwise kind of 😕 so I'm in the middle of disassembling the 8051 firmware with jhoenicke, and trying to squeeze what little bit of real performance can be had out of the device. I am targetting being able to do 25ish MHz single channel and 15 MHz dual channel _with actual triggering_ by shipping all of the data immediately asynchronously back to the host and letting the host figure out the trigger in parallel with data collection.
>
> An open source firmware would be very cool.
>
> Do you have information/pointers on the firmware code and packaging?
> I would love to at least follow the project even if I might not have time to actually hack on this (my 6022BE device is enroute so no real hacking besides disassembly for me for now).
>
> I recently wrote firmware for an existing power supply (B3603) and it was kinda fun to do that.

So firmware isn't so bad on this device. If you look at my repo, it's got the hex code for the stock firmware; the device has no sizeable non-volatile storage to speak of, so the original Windows driver sends the firmware over USB every time the device is plugged in. I was able to capture the data by running Windows in a VM and running a trace on the USB port in wireshark. The MCU that controls the device in the scope is a CY7C68013A (FX2LP), which is really ubiqitous. It sports a lot of GPIO, an 8051 processor and a 4K FIFO that buffers the data. The ADC is an AD9288, which is also as common as dirt. If you're interested in the firmware, i hope you like assembly 😆. Cypress kind of suggests you use Kiel but I am using mcu8051ide, and there are lots of open source 8051 tools including disassemblers out there; the instruction set on the device is also pretty sane. Pretty much everything use in the scope is well documented, and I can provide further details if you have questions.

Reply Quote Notify

6 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** April 28, 2015, 01:50:06 am »

Quote from: daybyter on April 25, 2015, 08:57:29 pm

> Phew...python skills....I sorta know how it works, but I don't really like it. I'm more a java guy. Your C rewrite might be easier to read for me.
>
> You think, you can get that many data across USB with a constant data rate? Might be some mouse movement come in between, or so...?

Yeah, there's gonna be some tricky code in there for the time being because we're getting close the limits of performance on CPython (and threading is unfortunately sort of nerfed in CPython anyways which makes life even harder). I am going to try and roll a new oscilloscope program, so you might just watch out for that; I am debating how much if any needs to be done in C (really need a high speed ring buffer).

As far as how much data can be moved, the big thing is you have to make absolutely sure the device is on a bus by itself. You can do this by running lsusb on Linux (not sure what the equivalent for Mac or Windows is), and checking to make sure no other devices on the same bus. As far as moving data, USB 2.0 kind of caps out 30 MB/s so that really sets an upper limit. With the help of Jochen Hoenicke, we've already changed the firmware and driver in a few critical ways that make it significantly more performant than what is available stock. First, Jochen has removed dead code from the firmware, and added an option to only pull data from one channel (by default it always pulls both); this allows you to expend all your USB bandwidth getting more samples for a single channel. Jochen has also added more sampling modes; the default maximum was 48 MSa/s which isn't realizable with USB 2.0 and produces super dirty signals. Jochen has added a new 30 MSa/s mode, which in my testing is way cleaner, and I have been able to **stream at 30 MSa/s with less than 2% data loss**. On the driver side, I pulled a lot of the stupid Hantek behavior out; the original Hantek driver will try to clear the FIFO on every read, which means you always will lose data, and waste USB bandwidth. The stock driver also did everything synchronously and did not queue up bulk transfers, leading to more data loss and poorer use of bandwidth. I've got an async implementation that makes these changes, and I can get really looking traces at reasonably high speed (30 MSa/s); ideally with a better C implementation or some clever use of multiprocessing, I will be able to realize this performance in an application.

What does this mean? You'll probably not get the advertised "20 MHz" performance exactly, but you'll probably get 5-7+ MHz single channel (or better, with good interpolation), and probably 3-5+ MHz dual channel, and hopefully get good consistent triggering. A lot of this is going to involve lots of interesting software magic. 😉

The board also appears to have traces and holes for an external trigger, I am still examining what it will take to get this integrated into the firmware.

Reply Quote Notify

7 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** April 24, 2015, 09:15:18 pm »

Quote from: daybyter on April 22, 2015, 09:54:18 pm

> I'm trying to run your code in a virtualbox vm, but no luck to far. Kernel shows the scope as an unknown usb device with the correct id, but it seems the python code cannot find the device.
>
> Has anyone managed to run this in a vm?

I have not tried running it in a Linux VM, but I could try later. Can you post a screenshot and/or open a ticket on Github? Any error output in general dumped from the console? The code is still a little rough, how good are your Python skills?

As a more general update, it turns out the firmware Hantek shipped this device with has got some bugs and is otherwise kind of 😕 so I'm in the middle of disassembling the 8051 firmware with jhoenicke, and trying to squeeze what little bit of real performance can be had out of the device. I am targetting being able to do 15ish MHz single channel and 10 MHz dual channel _with actual triggering_ by shipping all of the data immediately asynchronously back to the host and letting the host figure out the trigger in parallel with data collection. I'm also going to try and get the Python library rewritten in C and make it compatible with all three major OSes (since libusb should be present everywhere). The board also is provisioned for an external trigger, which appears to be connected to a GPIO on the FX2LP, which may prove fruitful. The end goal is to be able to write some open source firmware for this scope and at least provide the means (via a library) to have this device function in some sort of meaningful way, like having a working trigger (even if it has to be done host side). I have not decided how much effort I am going to put into rolling a GUI for the library, but it will eventually be more than enough for someone reasonable versed in Python to roll their own. Though this thing is sort of underwhelming in comparison to even my Rigol 1102E, it might be an interesting tool for the Raspberry Pi audience to be able to debug their low speed Arduino and what not via their Pi for under $100 (provided a few hardware hacks and some generous software magic is applied).

Reply Quote Notify

8 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** April 20, 2015, 07:59:08 am »

Quote from: evgenymagata on April 13, 2015, 03:17:27 pm

> Hello,
>
> first of all, thanks to the people involved with the Open6022BE Software. I got a question though: Since i only have Mac and Linux computers, i am forced to run the Software in a Windows XP Installation in a VirtualBox VM on my Mac.
>
> Generally, the Software works great. Unfortunately for some reason VM seems to limit the options of the DSO significantly: in the horizontal DIV i can only select time ranges between 10us and 2ms or the program crashes immediately. This happens to both the original Hantek and the Open6022be software. Does anyone have an idea how i could resolve this problem?
>
> Cheers,
> e.

I had worked on this little cheap-o scope a while back, and got the weird inclination to sit down and reverse engineer it this weekend, so I wrote some Linux (and probably also Mac OSX) bindings for Python (via libusb) if you were looking to use this on a non-Windows machine: https://github.com/rpcope1/Hantek6022API

Reply Quote Notify

9 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** January 20, 2014, 04:32:31 pm »

Quote from: Matchless on January 19, 2014, 04:37:16 pm

> Hi I joined this forum after reading this great thread. I have also bought the Hantek 6022be for all the reasons already mentioned. The software RichardK is busy with just made it more promising. I still need to understand how to use this DSO properly, as this is all new to me.
>
> I will try to post the noise shown on both my channels when connected to the internal square wave. It interesting to see that channel 2 is nearly clean compared to channel 1.
> Both probes were set to X10.
> Any comments please?

Matchless,
Have you added any shielding internal to the oscilloscope? I followed some of RichardK's instructions (thanks again so much dude), and added a couple extra caps and some shielding to the DC/DC converter, this should reduce a lot of the noise you're seeing. I also ran a power spectrum on the base level noise coming out of the oscilloscope after the fact (if you look back a couple pages) and found that most of my noise now was 60 Hz noise, which just means the whole thing needs better shielding (the oscilloscope is plugged into a laptop, so I was pretty sure that probably isn't a noise source). I wouldn't be totally surprised a stock oscilloscope was more noisy on channel 1 than 2, maybe if nothing else because of channel 1's proximity to the DC/DC converter, which really according to it's spec sheet is a little to noisy to normally be used in a scope. Can you build a power spectrum from that signal (and crop out the 10 kHz and harmonics), and show us what the noise looks like? I can certainly say from experience if you add the shielding and caps RichardK graciously posted about you should be able to get the noise about as low as it's going to go for this device. Also, I can post the python script I use to generate power spectrums from the Hantek wave output files if that would help.

RichardK,
I started to use your binary. Though I can tell it's still a work in progress, this looks so much better. Thanks for your hard work! 🙂

Reply Quote Notify

10 Repair / Tektronix PS280 Repair

« **on:** January 04, 2014, 07:49:26 pm »

Hello everyone! I've got a Tektronix PS280 power supply I bought a while back at a surplus auction which has got a channel broken. This is a 2 adjustable channel + 1 5V rail power supply, so I've been ok just using one channel, but I would like to try to figure out how to fix it. The second channel will report close to 44V (this is a 0-30V PS) if you bring the current control above near zero, with absolutely no load on it. It does not appear to affect any of the other channels, so I usually just leave the current turned all the way down and ignore it. The only thing I've ever seen on the internet even addressing this problem is http://www.electronicspoint.com/tektronix-ps280-troubleshooting-problem-channel-output-t70457.html , which sounds a lot like my problem, however (this has been a couple of years), I think I tried to take the power transistor out and didn't notice anything funny about it. If it helps, I can also open it up and take pictures. I think this power supply is actually a rebadged GW Instek but I'm not sure which. Any help or input is highly appreciated! 🙂

Reply Quote Notify

11 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** December 31, 2013, 11:44:43 pm »

Quote from: RichardK on December 30, 2013, 06:27:46 am

> Little bit more progress on the software side (see attachment)...
>
> I have implemented two different cursor modes, one exactly like the cursor mode in the original software (Cross, Horizontal, Vertical) and one I call Interactive, where two cursors labeled 1 & 2 in little circles are drawn with the waveform and you can interact with

> them by dragging them across the waveform, and they will follow the wave's y axis.
>
> By the way, ignore the ugly toolbar button glyphs, those will not be in the final version as I am going to make my own graphics for the toolbar but those are placeholders until I am done.
>
> Anyway, lots more work to do on it, but thought I'd share my progress thus far for those who are interested, thanks 🙂

Richard, this looks awesome. Thanks for sharing! 🙂

Reply Quote Notify

12 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** December 23, 2013, 02:43:41 am »

I attached a *background noise* power spectrum for my now modified 6022BE, which I think is pretty revealing. This scope is attached to my old laptop. When I remove the charger cable, the noise seems to grow slightly. I guess this thing doesn't have sufficient enough shielding or whatever else to reject all of the regular EMI at this level. Still I get flickers of noise at 3.8 mV; I suspect this only a single bit on the DAC, and I'm not sure we're going to do much better for noise reduction on this device. Still I'd be open to ideas to reducing it. 😀

Quote from: rpcope1 on December 13, 2013, 03:13:22 am

> **Quote from: RichardK on December 09, 2013, 07:01:22 pm**
>
>> Here are the inside pictures, using a cell phone camera at the moment so they are not very bright or crisp...
>>
>> As you can see, I have already started populating the alternative DC-DC section with SMD Capacitors, and I noticed some Hantek equipment has the Crystal can grounded so I did that as well, probably doesn't change much but it didn't hurt it either.
>
> Wow! This looks pretty good. I'm going to try to do the same for mine. Thanks so much for posting your pictures and keeping us updated! 🙂

Edit: Clarified what the power spectrum was referring to.

Reply Quote Notify

13 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** December 23, 2013, 02:13:05 am »

Hey guys,

Just to build on RichardK's work, I got around modifying the hardware on my 6022BE this weekend. I added the 100 uF capacitors to the +5V USB rail and the +3.3V rail, and made an aluminum foil shield for the DC-DC converter, which was soldered to ground. I also added film 1 uF capacitors between +V0 and 0V and between -V0 and 0V. These capacitors hopefully don't push the DC-DC converter too far as capacitive load goes. It looks like I'm down to some sporadic 2-3 mV noise, which given how much I've got in this, isn't bad. RichardK, I really like your new GUI; I think C++ is a much better choice for this than Python, if nothing else for the speed. Thanks again for your hard work. 🙂

Edit: Spelling error.

Quote from: RichardK on December 13, 2013, 07:24:08 pm

> No problem guys, my pleasure.
>
> **Quote from: Mark_O on December 12, 2013, 04:03:03 pm**
>
>> Lastly, you enumerated 6 tweaks you made to the board, though I thought I counted more like 8. But in any event, it would be helpful to know the relative contribution of each, to maximize 'bang for the buck'. Not necessarily to minimize cost, but perhaps time. I suspect, and it's highly probably, that some tweaks will have very minimal influence. And omitting them really wouldn't matter.

> Of course, to conduct such a study would require stopping to evaluate after each mod was installed. I realize that's asking a bit much. But just a thought for those following in your footsteps, if they'd like to make their own contributions.

> I don't want to hog all the fun 🙂

Reply Quote Notify

14 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** December 13, 2013, 03:13:22 am »

Quote from: RichardK on December 09, 2013, 07:01:22 pm

> Here are the inside pictures, using a cell phone camera at the moment so they are not very bright or crisp...
>
> As you can see, I have already started populating the alternative DC-DC section with SMD Capacitors, and I noticed some Hantek equipment has the Crystal can grounded so I did that as well, probably doesn't change much but it didn't hurt it either.

Wow! This looks pretty good. I'm going to try to do the same for mine. Thanks so much for posting your pictures and keeping us updated! 🙂

Reply Quote Notify

15 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** December 09, 2013, 07:28:22 am »

Quote from: RichardK on November 28, 2013, 09:36:08 pm

> I have reverse engineered the Front End if anyone is interested:
>
> The A7 device is just a fast switching rectifier, it breaks down at 100V and it's being used to clamp the signal to -5V and +5V.
>
> Also take note that the instead of your usual 1M ohm resistor and trimmer capacitor in parallel connected from input signal to ground we have the input signal going through a 909K resistor with the trimmer capacitor in parallel with it, then shunted to ground through a 100K resistor and an SMD capacitor in parallel. To the input, it looks like a 1M resistor to ground, but the signal is being tapped between 909K and 100K, then going to the first Op Amp. I'm not sure a 5V input signal in 1x mode is going to give a 5V signal at the node between the 909K and 100K, but more like 500mv. This would mean the probe in 1x mode is going to be safe all the way up to 50V and 500V in 10x.
>
> Also, the outside of my unit shows a label between the BNC connecors that says "35vpk max" which I am not sure if they mean for 1x or 10x?

Richard,

I would be very interested in what you've done with the front end. I wrapped the SDK into Python via ctypes, thinking I would use this scope for data acquisition and possibly doing some spectrum analysis. To you and anyone else that might be interested, I uploaded my Python wrapper to https://github.com/rpcope1/Hantek6022API . I hope someone can get something useful out of it.

Also to you and everyone else, I saw someone mentioned changing or fixing the DC-DC converter. I can see where the Mornsun DC-DC convert is soldered in (and I agree, it's a terrible choice for this application). I think the first thing I'm going to attempt to do is solder some big electrolytic caps onto between both the V+ and V- and the 0V reference coming out of the converter. Does this sound reasonable thing to do to reduce noise on the traces? I'm also open to changing the DC-DC converter here if anyone has some suggestions as to how to do, and reporting how it worked. I think this will make an interesting experiment.

Also, I'm interested in any other experimentation you all are thinking about (including possibly hacking the firmware?).

Reply Quote Notify

16 **Test Equipment / Re: Hantek 6022BE 20MHz USB DSO**
« **on:** November 25, 2013, 06:59:43 pm »

Hello everyone,

I've been watching this thread for a while, and I just wanted to say thanks for putting up all the cool information. I bought this scope a few months ago, thinking I'd do like what Aurora was talking about and mod it out. I've been writing a Python front-end for the API Hantek provides (I'm not huge on the provided scope software), and was thinking about doing the same in C++. I saw the OpenHantek project but it doesn't look like it supports the 6022BE scope. Is this something you all might be interested in? My big issue with the scope right now is the noise level. In reading this, I thought I read that a DC/DC converter was causing the roughly 10 mV noise seen in the channels? Do you all think it's possible to clean up the noise and get slightly better resolution as far as the signal goes, or is that asking too much from this scope? Maybe I just need to pony up and buy a dedicated spectrum analyzer (that's what I was hoping to accomplish). 🙂

Reply Quote Notify

Pages: [**1**]

EEVblog Main Site EEVblog on Youtube EEVblog on Twitter EEVblog on Facebook EEVblog on Odysee