Siglent SDS1104X-E Review

input frequency and *fs* is the sample clock, whereas *n* and *m* are just integers running from 0 to (theoretically) infinity. During normal operation, we don't want to see any mixer products, which is perfectly possible as long as the input signal and all its harmonics don't exceed *fs*/2 and the output of the ADC has a brick-wall filter (then in the digital domain of course) that removes everything above *fs*/2. But in some circumstances, we can make use of a certain high-order mixer product, just as in this example, where the effective FFT sample rate is only 2MSa/s, which is obviously much too low for an 81.4MHz signal.

According to the formula given above, we are aiming at the mixer product for -1 x *fi* + 41 x *fs*, which is

41 x 2MHz – 1 x 81.4MHz = 82MHz – 81.4MHz = 600kHz;

Now if we set the center frequency to 600kHz, we get the carrier at 81.4MHz and can also clearly see the sidebands 400Hz apart from it. 16x Averaging has been used in order to get a clean and stable display.

There are several drawbacks though:

- With this mixing scheme, we get the result in reverse frequency position, i.e. the upper sideband appears below the carrier and vice versa.
- Mixing with the 41st harmonic of the sample clock introduces also 41 times more phase noise and jitter and it clearly shows in the FFT plot. Just look at the filter shape of the individual spectral lines.
- Amplitude accuracy is pretty much gone, as a harmonic mixing process cannot be as efficient as the fundamental one, hence we get about 4.6dB attenuation. Note that I had to reduce the input level by 3dB compared to the first screenshot in order to avoid input overloading, so 3dB of the total difference are caused by that and not by the measurement error due to harmonic mixing.

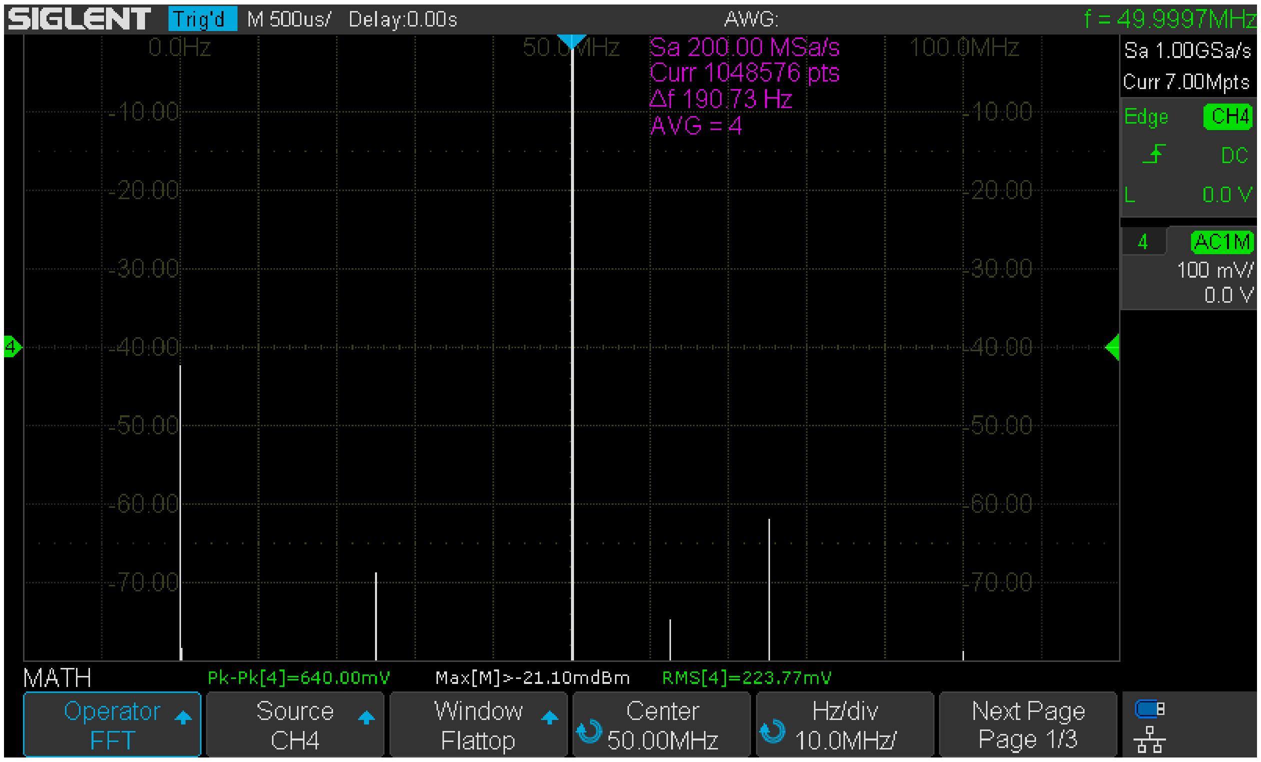
So this is not an ideal application, just some emergency measure to get a result – following the motto "a compromised measurement is better than nothing at all …"

Performance Test

Up to now, we have just explored the possibilities to properly set up an optimal FFT analysis for various tasks, but we have not checked the actual performance of the FFT implementation in the SDS1104X-E. We can often hear opinions suggesting that the FFT length is the most important factor and if this were true, the Siglent SDS1kX-E series with 1Mpts max. FFT length would be hard to beat. Actually, FFT length only determines the frequency resolution and noise. Yet for the majority of tasks, like characterizing an unknown signal and interference hunting, we don't really need an extremely high frequency resolution. Likewise, for the vast majority of measurements with a DSO, noise isn't the limiting factor either. Consequently, a long FFT is nice to have, but most of the time a high spurious-free dynamic range and low harmonic and intermodulation distortions would be much more important. For this, an 8-bit sampling system sets tight limits with about 49dB dynamic range, which cannot be changed by increasing the FFT length or any other averaging techniques. We can indeed get more than that in certain scenarios and an e.g. 70dB dynamic could easily be demonstrated with a special setup, but this is not universally true and ironically fails just for the majority of real world applications. So while we should not expect any wonders, the FFT in this scope is still a very powerful implementation and very useful tool within the constraints of the 8-bit acquisition system.

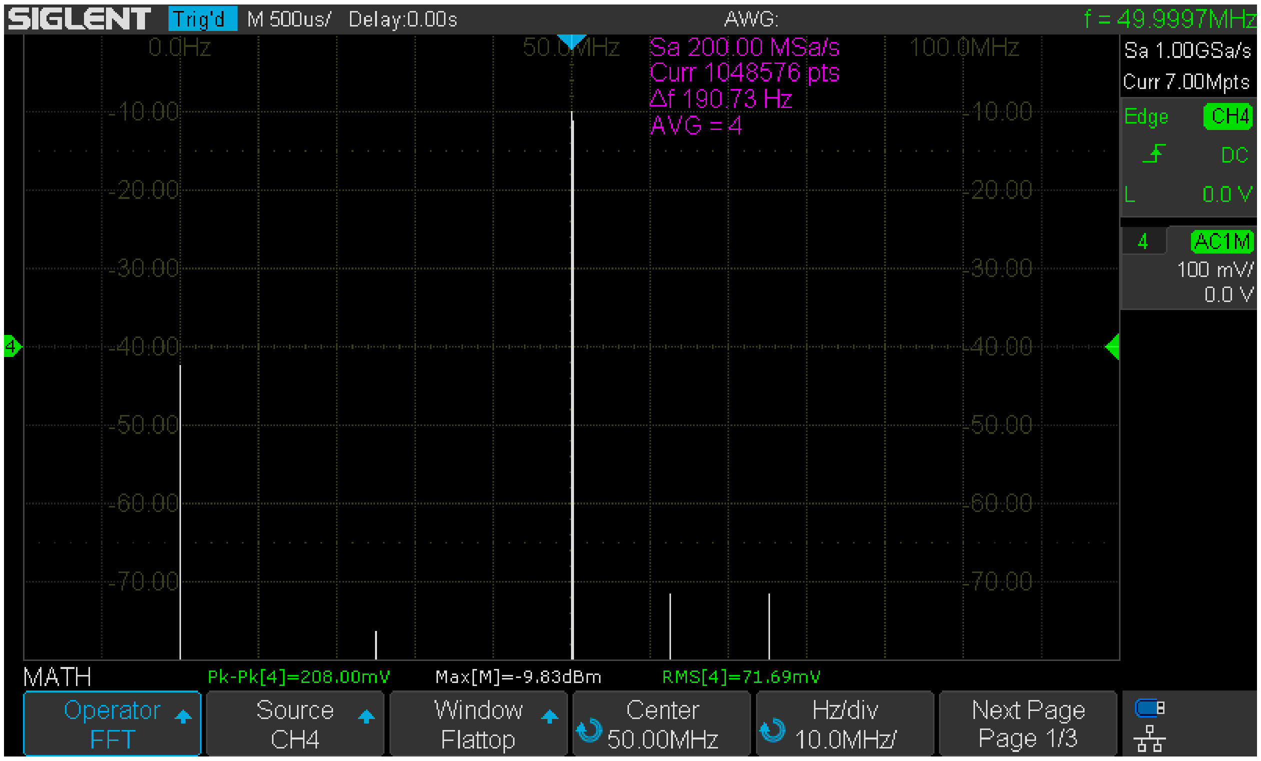
Amplitude Accuracy

Analyzing a signal in the frequency domain is about exploring its spectral components with their amplitudes and maybe also relative phases. Well, we obviously don't get any phase information and this is quite common in DSO-FFT implementations as well as scalar spectrum analyzers. But we do expect decent amplitude accuracy within the dynamic range of the 8-bit system, at least when using the Flattop window. For this test, a 50MHz sine from a DDS function generator is fed into channel 4 of the scope via a precision step attenuator and this combination is capable of providing a fairly accurate amplitude range of more than 120dB. Let's start with 0dBm using the low noise gain of 100mV/div and an analysis bandwidth of 100MHz so we can see all the spurious signals generated by the scope itself.



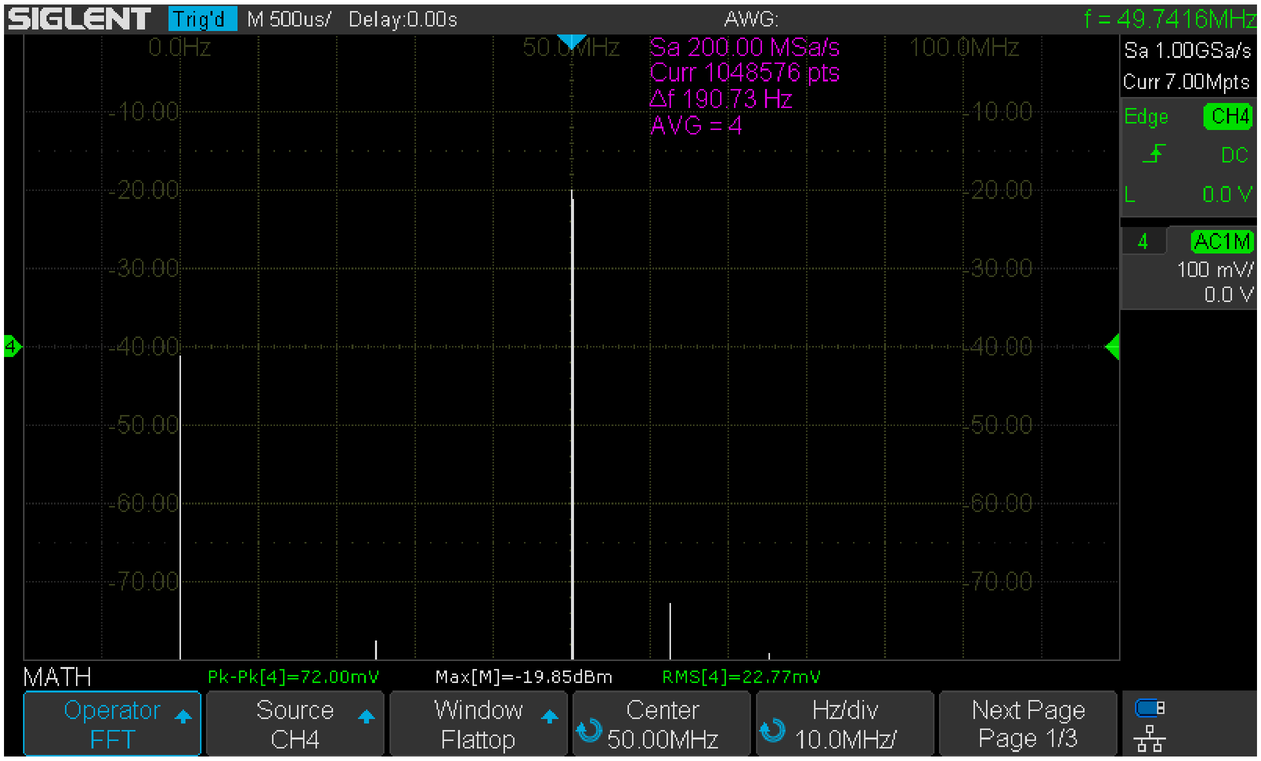
SDS1104X-E_FFT_100mV_Amp_0dBm

The signal strongest spur at 75MHz is -62dBm, hinting on a spurious free dynamic range of ~62dB.



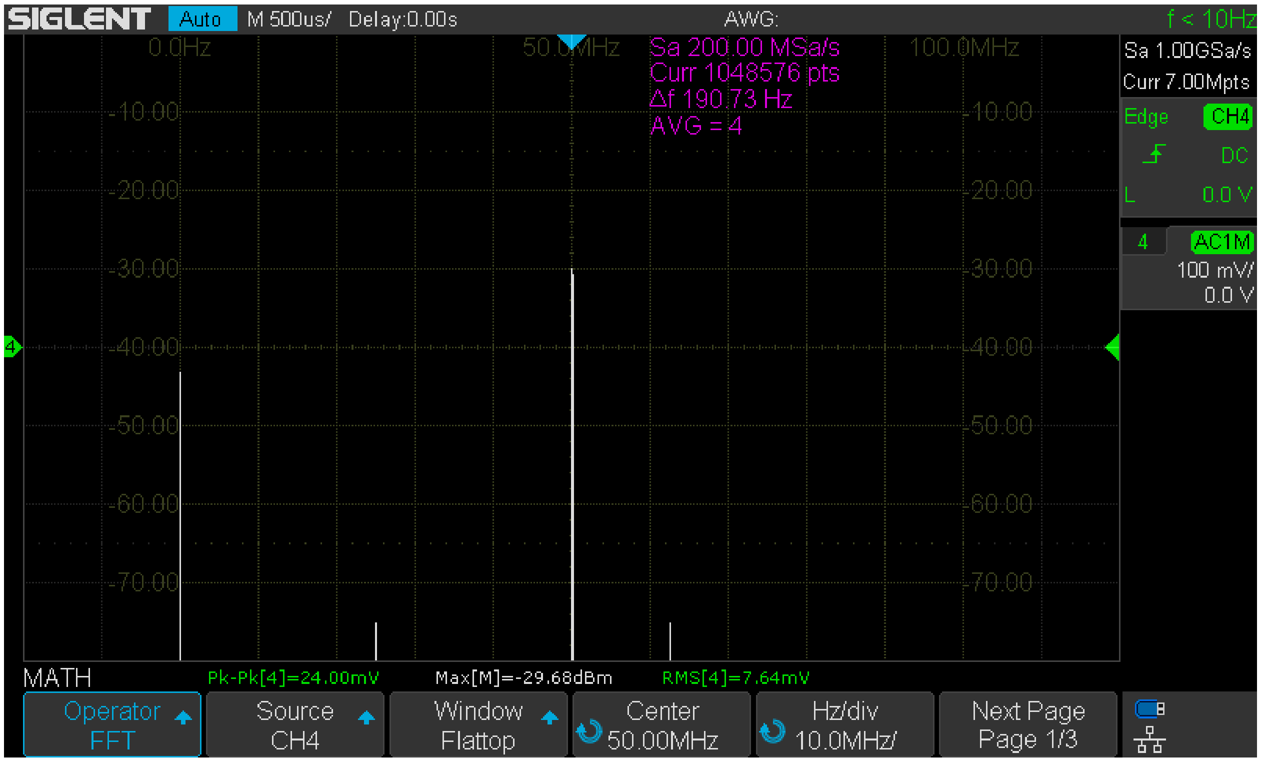
SDS1104X-E_FFT_100mV_Amp_-10dBm

With a signal level of -10dBm, the spurious signals at 25 and 75MHz have dropped by nearly 10dB as well, which indicates they are directly input signal related. In contrast, the spur at 62MHz is even stronger now, so it’s only loosely related to the input signal level. The next screenshot shows a -20dBm signal:



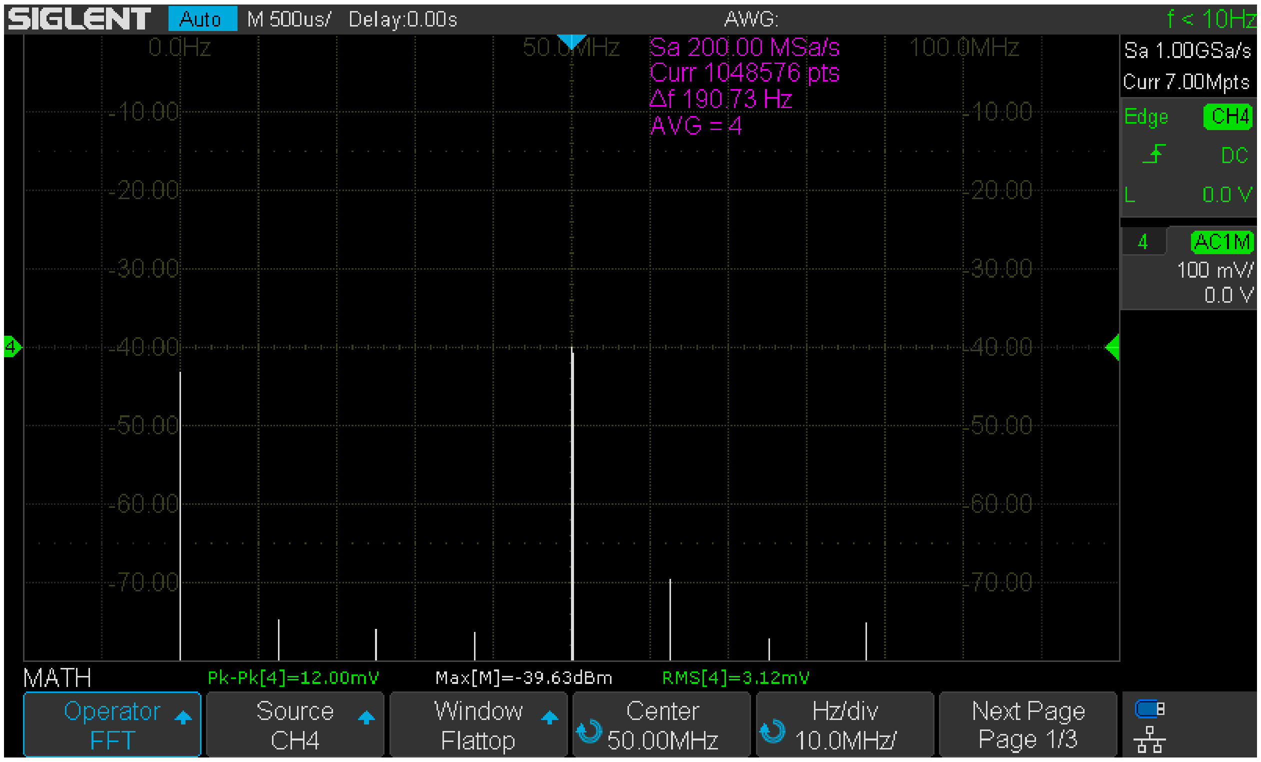
SDS1104X-E_FFT_100mV_Amp_-20dBm

The spurs at 25 and 75MHz have dropped even further, while the one at 62MHz remains fairly constant.



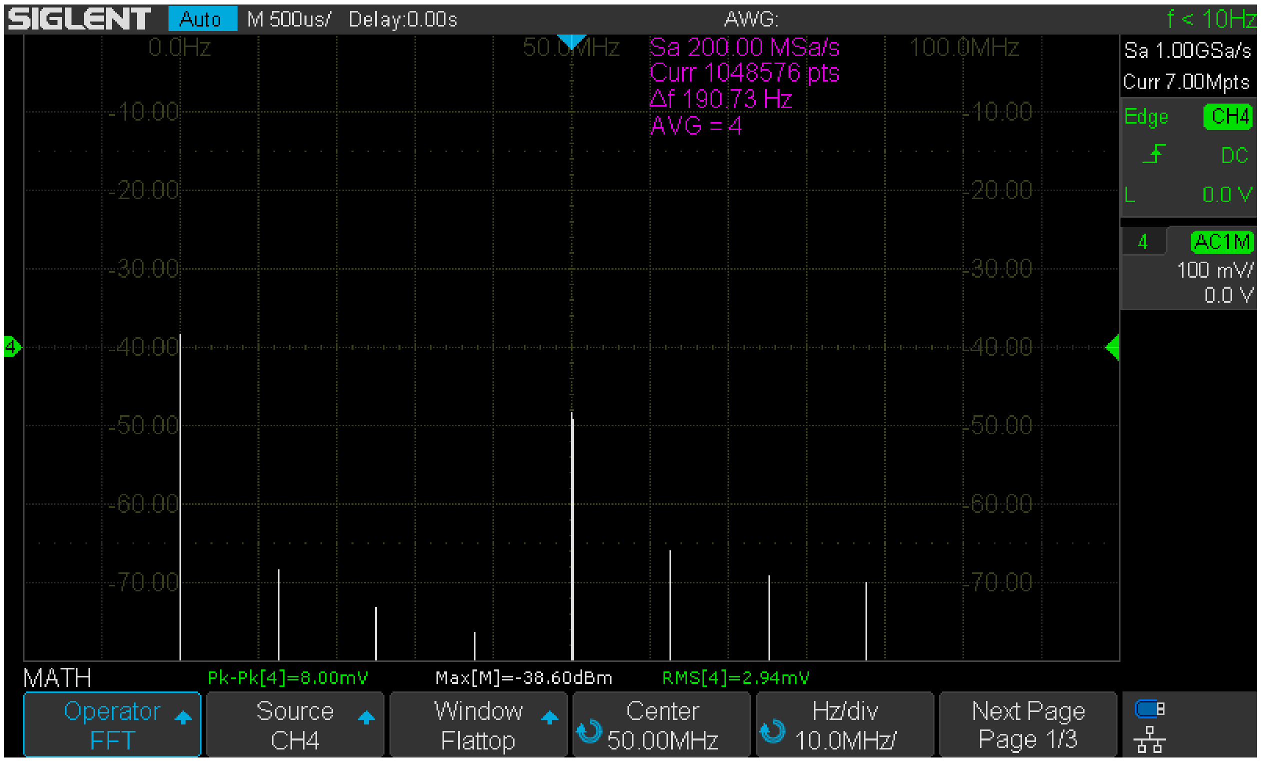
SDS1104X-E_FFT_100mV_Amp_-30dBm

At -30dBm the spur at 75MHz has completely vanished, but the other two remain at -75dBm.



SDS1104X-E_FFT_100mV_Amp_-40dBm

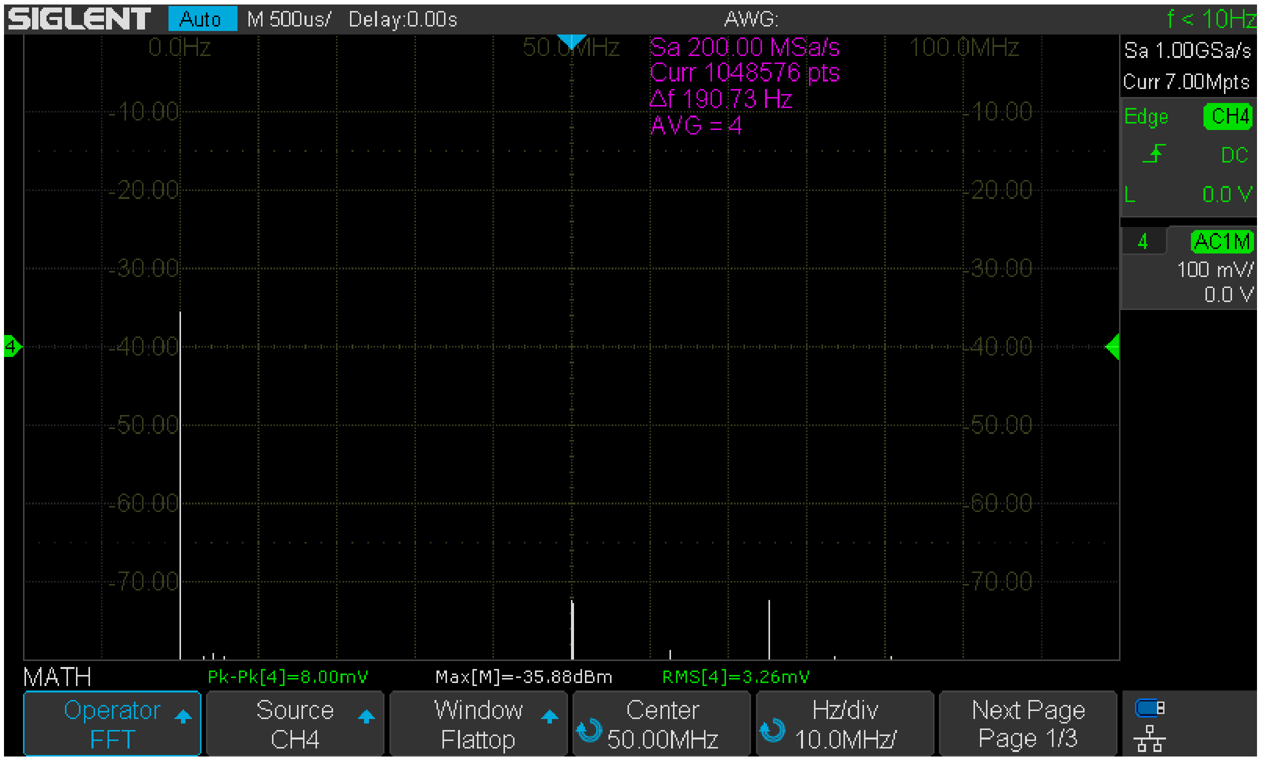
At -40dBm, there are only a couple ADC LSBs used anymore and linearity gets worse. A new maximum is reached at 62MHz and several other spurs have emerged.



SDS1104X-E_FFT_100mV_Amp_-50dBm

Up to this point, the Max measurement on the FFT has been pretty accurate, but now it catches on the DC component and has therefore become meaningless. Other than that, the signal level is displayed as some

-48dBm, so we're starting to lose accuracy as we're finally approaching the limit of the 8-bit dynamic range.



Sa 200.00 MSa/s
Curr 1048576 pts
Δf 190.73 Hz
AVG = 4

Pk-Pk[4]=8.00mV Max[M]=-35.88dBm RMS[4]=3.26mV

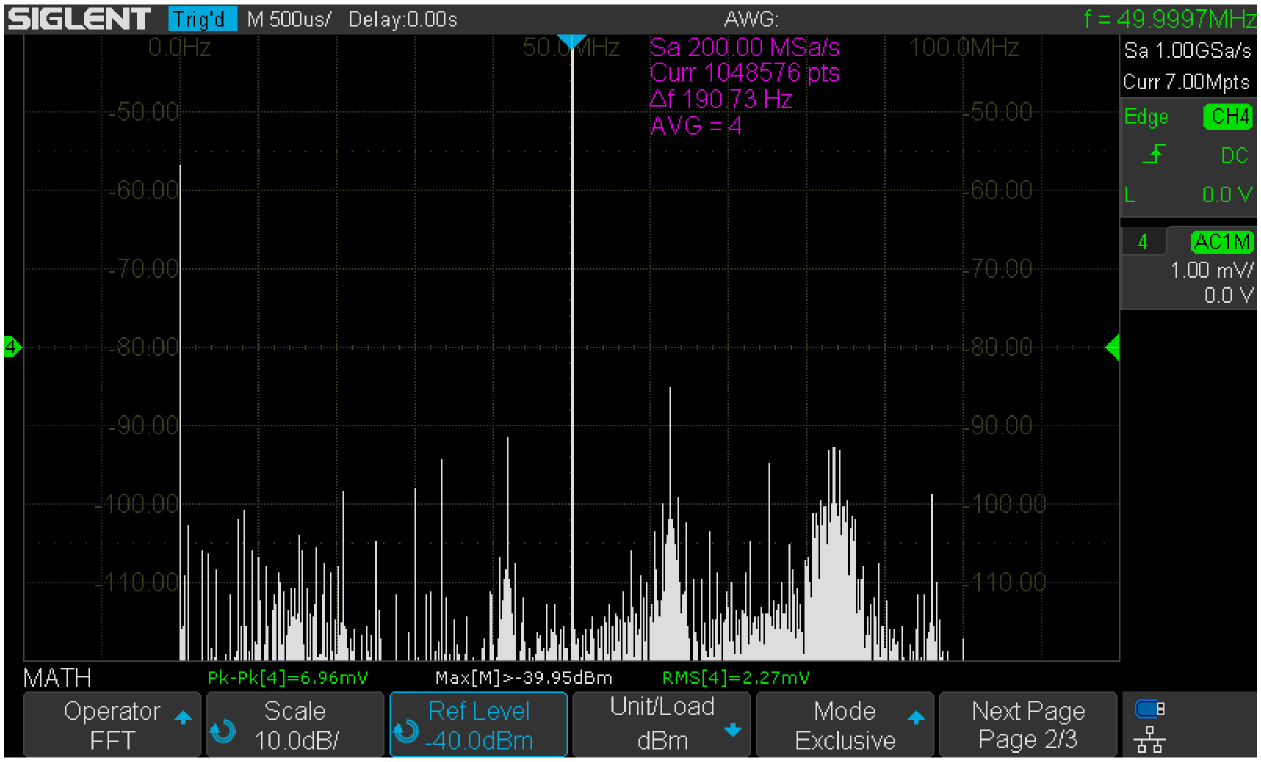
SDS1104X-E_FFT_100mV_Amp_-60dBm

At -60dBm, the spectral line for the signal drops dramatically, showing some -73dBm. Even though there is no visible noise and only one substantial spur, the range below -50dBm is simply unusable for single signal measurements due to the constraints of the 8-bit sampling system. So don't make the mistake to think you have 80dB+ dynamic range, just because the noise floor appears so low and only few spurious signals are visible – which by the way comes as no surprise in this scenario, because all harmonics related to the 50MHz signal are outside the analysis bandwidth.

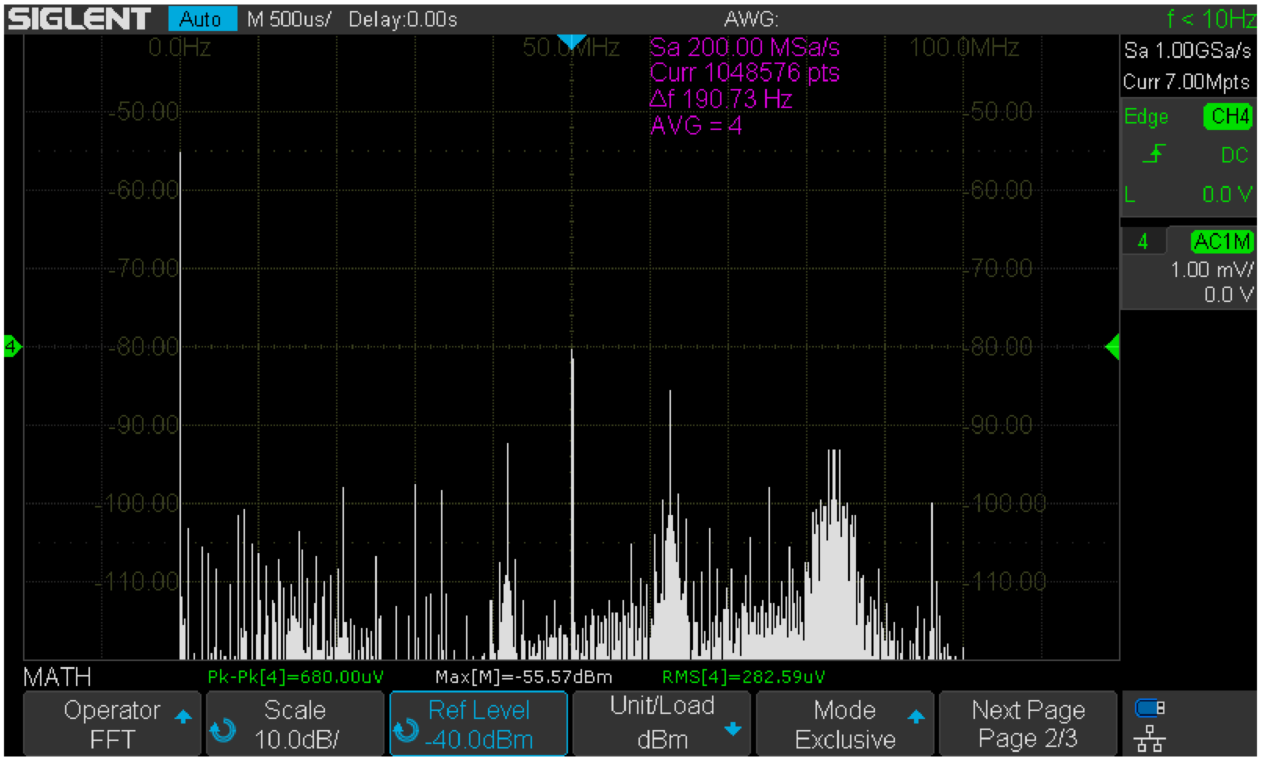
One might wonder how low we can go with the amplitude measurements. Up to now, we could demonstrate that it works fairly well down to -50dBm, which is just 707µVrms or 2mVpp. This is almost as sensitive as most AF or RF Millivolt meters with the added benefit of the selective measurement that ignores harmonics as well as most of the noise. But we can do better…

Up to this point we have been using 100mV/div gain throughout this test and could demonstrate accurate measurements down below one mV. By increasing the channel gain to 1mV/div there should be a boost in sensitivity by 40dB and we can hope to measure signal levels down to -90dBm, equivalent to 7µVrms!

The screenshot below shows the 50MHz signal at -40dBm. Due to the high channel gain, the noise level is down below -110dBm, but there are lots of spurious signals, not input signal related, just electrical noise and interference from inside the scope itself. The level of this is pretty low and there is certainly no reason to complain about spurs below -80dBm (<22µVrms), especially not for a cheap entry level DSO like the SDS1104X-E. Yet this ultimately limits the ability to measure unknown weak signals to about 10µVrms whereas for known signals like in this test we would be able to go even lower and the proposed -90dBm should be realistic.

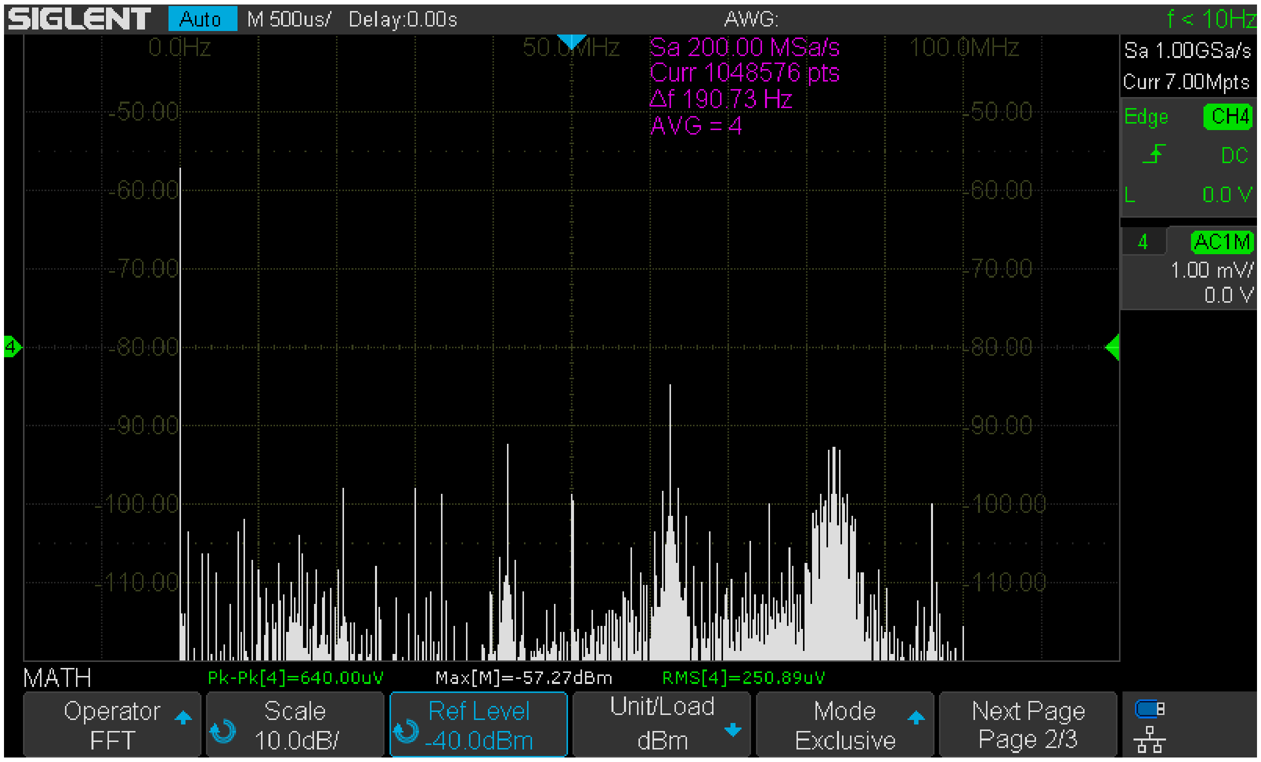


SDS1104X-E_FFT_1mV_Amp_-40dBm

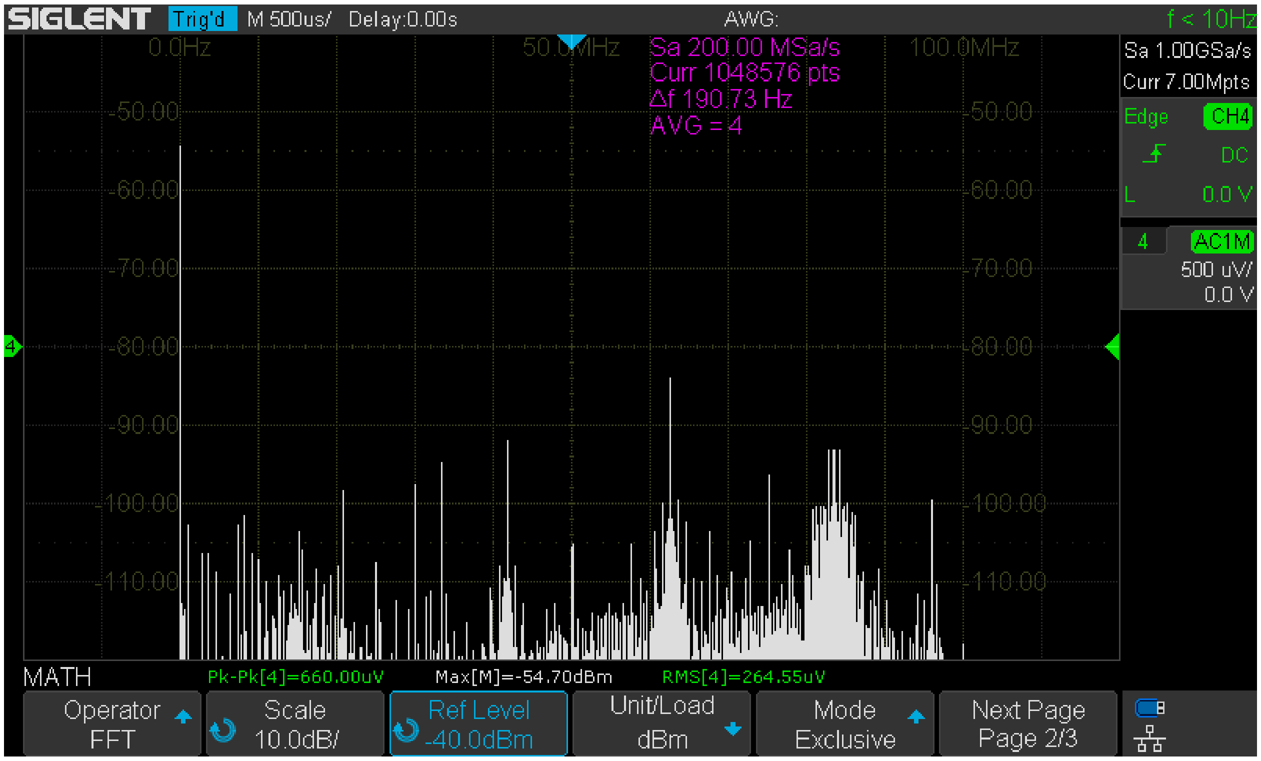


SDS1104X-E_FFT_1mV_Amp_-80dBm

At -80dBm as shown above, the automatic Max measurement has become useless again because of the DC component, but the spectral line for the signal still shows a pretty good level accuracy. As expected, all the spurs remain unchanged as they are all generated internally by the scope itself.



SDS1104X-E_FFT_1mV_Amp_-100dBm



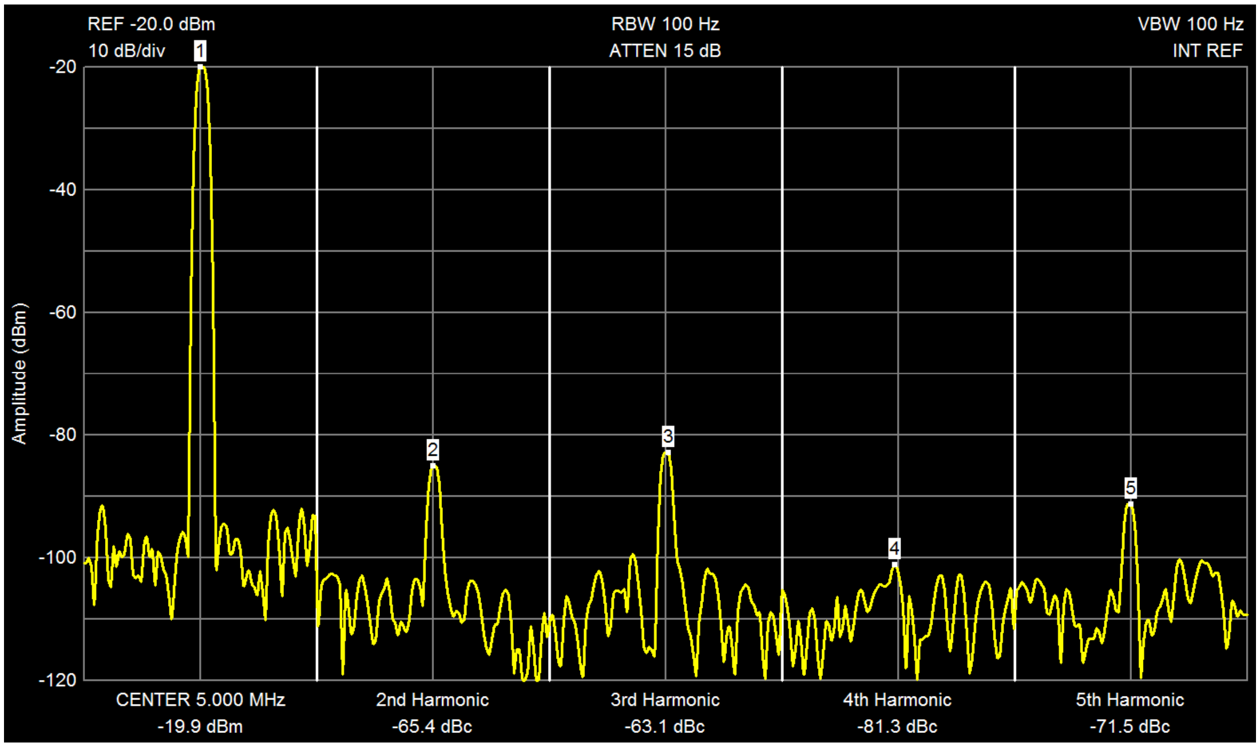
SDS1104X-E_FFT_500uV_Amp_-110dBm

As can be seen in the screenshots above, we can even measure the level of a known signal at -100dBm, that is 2,24µVrms or 6,32µVpp. This is the absolute limit though and the test fails at -110dBm, where the magnitude of the displayed spectral line is off by several dB even with the maximum channel gain of 500µV/div.

Harmonic Distortion

Another common task for frequency domain analysis is measuring the harmonics of a signal. Quite obviously, the harmonic distortion generated within the analyzer is a limiting factor for such measurements, hence it should be fairly low. Once again we'll have to face the limitations of an 8-bit system and cannot expect any better than about -46dB, which would be equivalent to 0.5% THD.

I've checked the harmonic distortion of the SDS1104X-E for several frequencies from 1MHz to 30MHz. Of course this test requires a low distortion sine wave, so the quality of the signal source has to be verified beforehand. This is done by means of a "proper" SA utilizing its "Harmonic Viewer" application. Below is a screenshot exemplarily showing the result for the 5MHz test signal.



THD_Ref_5MHz

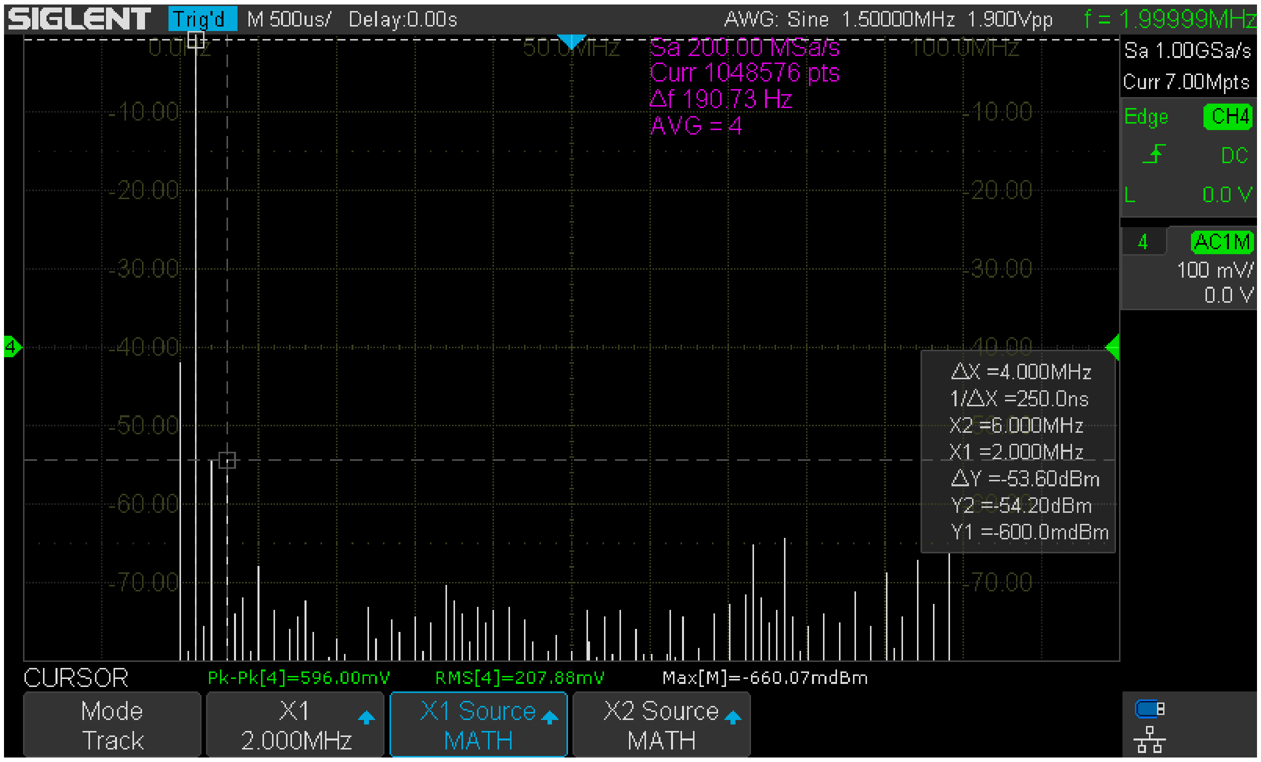
All the harmonics stay well below -60dBc at that frequency, which should be good enough for characterizing an 8-bit system.

Now let's see how the SDS1104X-E performs. The following screenshots demonstrate the harmonic distortion at 2, 5, 10 and 30MHz. Higher frequencies have not been tested, because the important 3rd harmonic would be outside the bandwidth of this scope. 1MHz has been tested as well, but the result was practically identical to the 2MHz test, so it has been omitted here.

No automatic measurements for the harmonics are available, but tracking cursors on the math trace are a near perfect tool to both measure and highlight (in the screenshots) the strongest harmonic.

As can be seen, the strongest harmonic is the 3rd and starting at about -53dBc at 2MHz it slowly degrades to -47dBc at 30MHz. All in all this is a fair bit better than expected and hints on a very good ADC linearity, i.e. its INL has to be better than 1LSB, even at high frequencies. This test would also reveal any problems within the frontend, but there is nothing to complain either. Please note the relatively low level of higher order harmonics and spurs!

The conclusion can only be, while this scope certainly isn't up to the task of characterizing low distortion sine waves, high fidelity audio gear or high performance RF circuits, it still performs pretty good for the majority of undemanding tasks with less strict requirements.

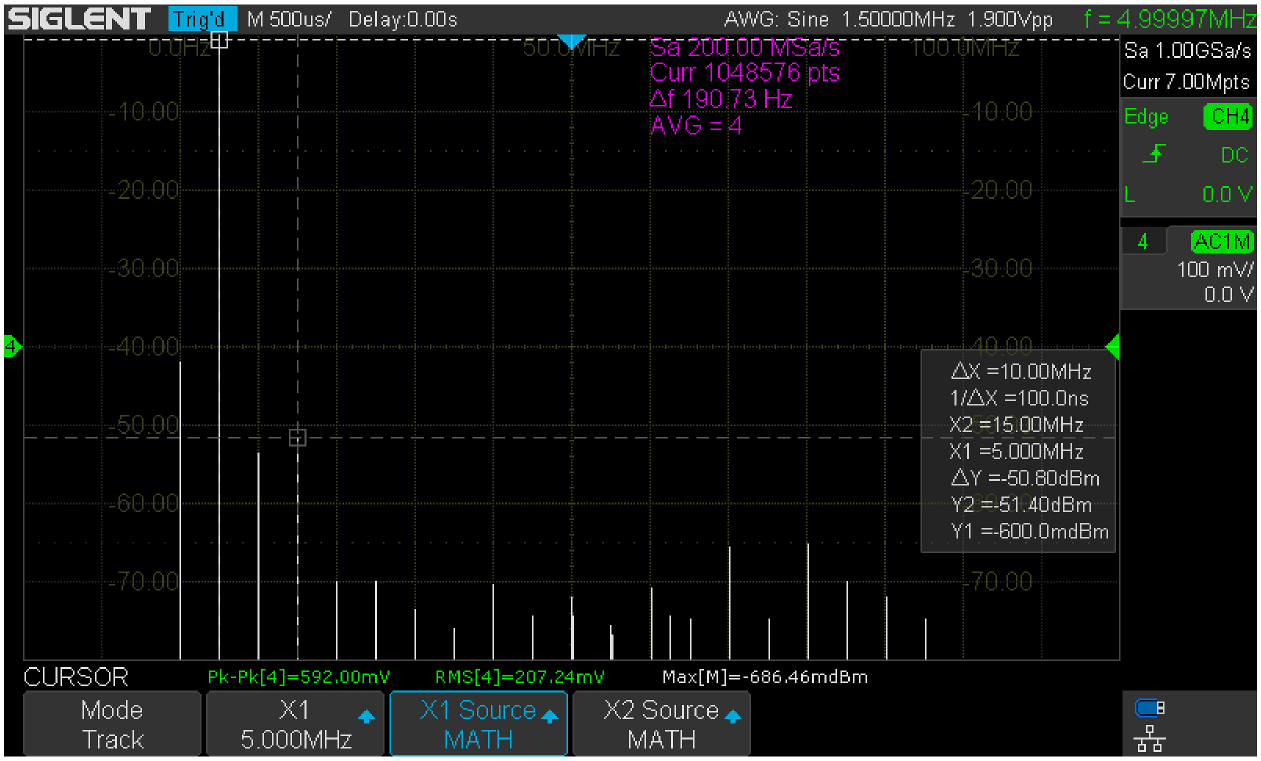


Sa 200.00 MSa/s
Curr 1048576 pts
Δf 190.73 Hz
AVG = 4

ΔX =4.000MHz
1/ΔX =250.0ns
X2 =6.000MHz
X1 =2.000MHz
ΔY =-53.60dBm
Y2 =-54.20dBm
Y1 =-600.0mdBm

Pk-Pk[4]=596.00mV RMS[4]=207.88mV Max[M]=-660.07mdBm

SDS1104X-E_FFT_THD_2MHz

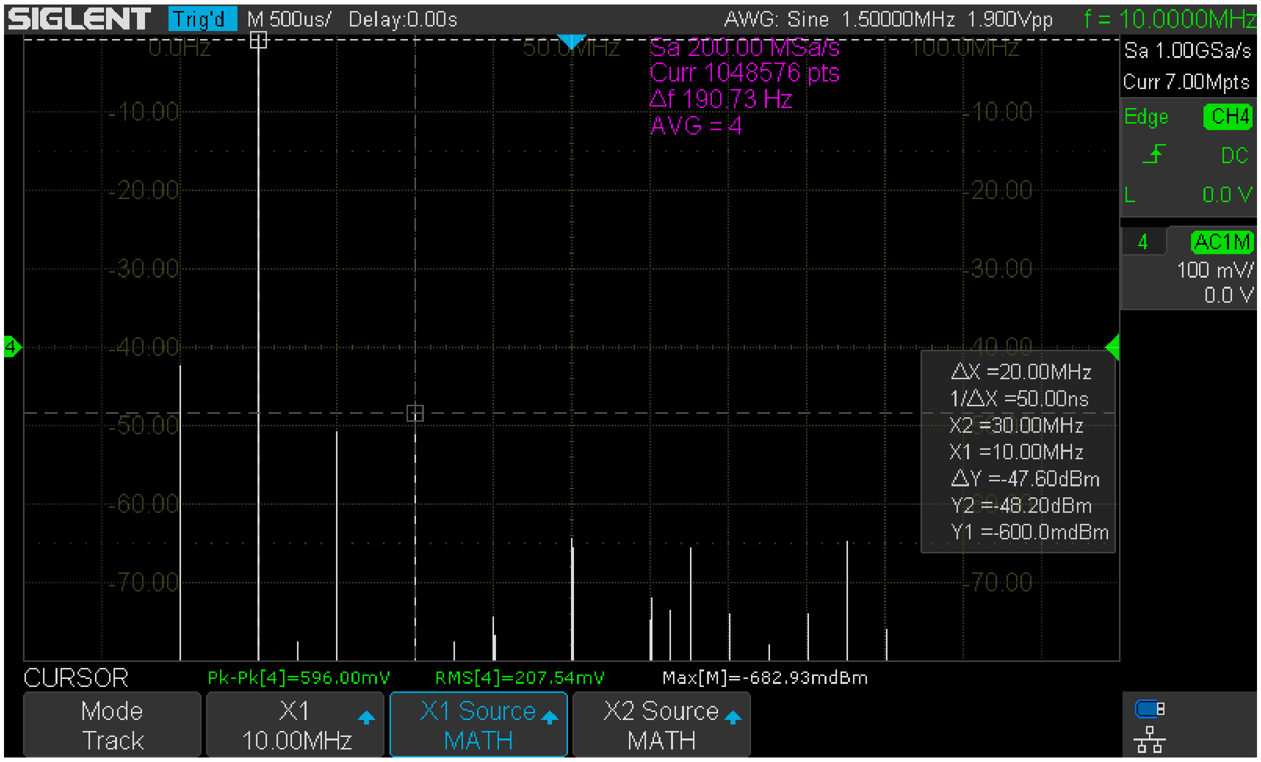


Sa 200.00 MSa/s
Curr 1048576 pts
Δf 190.73 Hz
AVG = 4

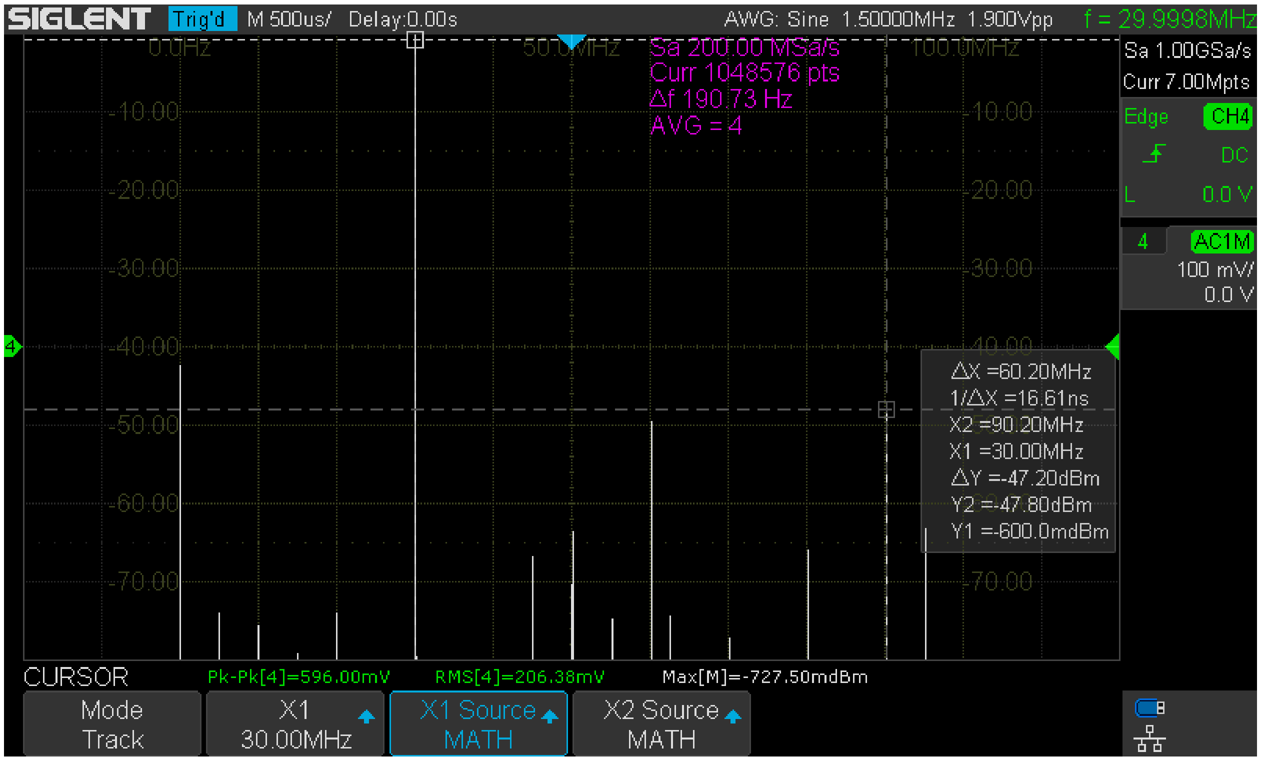
ΔX =10.00MHz
1/ΔX =100.0ns
X2 =15.00MHz
X1 =5.000MHz
ΔY =-50.80dBm
Y2 =-51.40dBm
Y1 =-600.0mdBm

Pk-Pk[4]=592.00mV RMS[4]=207.24mV Max[M]=-686.46mdBm

SDS1104X-E_FFT_THD_5MHz



SDS1104X-E_FFT_THD_10MHz



SDS1104X-E_FFT_THD_30MHz

Two Tone Test

The key specification for every spectrum analyzer in terms of practical use for narrowband measurements is its 3rd order dynamic range, usually expressed as IIP3 (3rd order Input Intermodulation Intercept point).

In short it's the ability to measure weak signals in presence of strong ones without generating unwanted 3rd order mixing products, which would appear near the original signals, thus creating a chaotic mix of wanted and unwanted (or better: relevant and irrelevant) signals – with the risk of unwanted signals even totally obscuring the real ones.

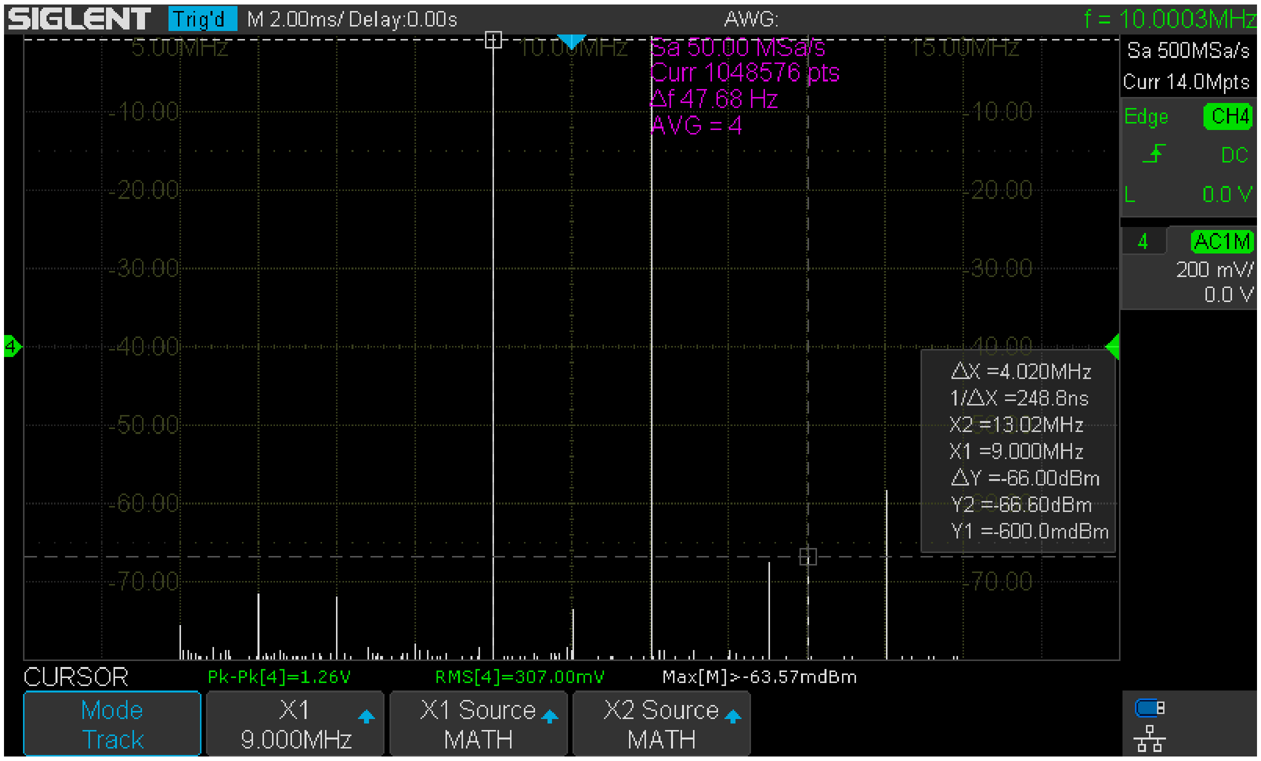
This test uses two input signals at 9 and 11MHz, both at a 0dBm level initially. The 9MHz signal is then attenuated step by step in order to check the amplitude accuracy of the measurements. At the same time, the 3rd order mixing products at 7 and 13MHz are measured, so the resulting IMD (intermodulation distortion) can be estimated.

The screenshots below show the results for 0, -20, -30, -60 and -70dBm.

The 0dBm test would be the classical setup for determining the 3rd order intercept point. The unwanted product at 7MHz is weaker at -72dB, but the opposite intermodulation product at 13MHz isn't strong either at only -66dB. This means a 3rd order intermodulation distortion of -66dB at 0dBm input level and the input intercept point would thus be a healthy +33dBm for a channel gain of 200mV/div.

At -20dBm, the intermodulation product at 7MHz vanishes (goes below -80dBm), the same is true for the 13MHz product at -30dBm. That's fairly decent and when looking at the other spurious signals, we can conclude that despite the 8-bit system, we can get at least 60dB dynamic range for narrowband applications like this.

We can also measure levels below -50dBc quite accurately, simply because the 2nd signal stays fixed at 0dBm and serves as a dither for the ADC. As can be seen, measuring -70dBc isn't a problem at all. Just for fun, I've added a measurement at -76dBm level, which still works surprisingly well.



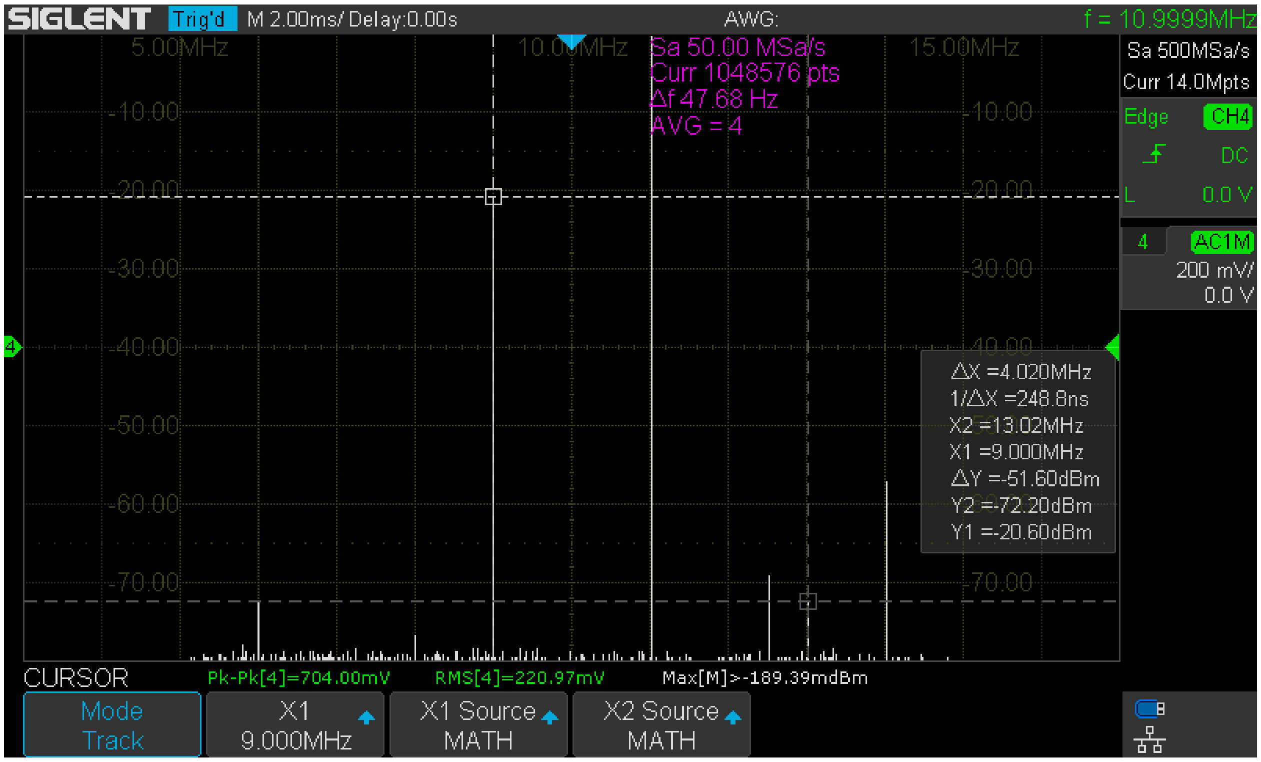
Sa 50.00 MSa/s
Curr 1048576 pts
Δf 47.68 Hz
AVG = 4

Sa 500MSa/s
Curr 14.0Mpts

ΔX =4.020MHz
1/ΔX =248.8ns
X2 =13.02MHz
X1 =9.000MHz
ΔY =-66.00dBm
Y2 =-66.60dBm
Y1 =-600.0mdBm

Pk-Pk[4]=1.26V RMS[4]=307.00mV Max[M]>-63.57mdBm

SDS1104X-E_FFT_IM3_0dBm

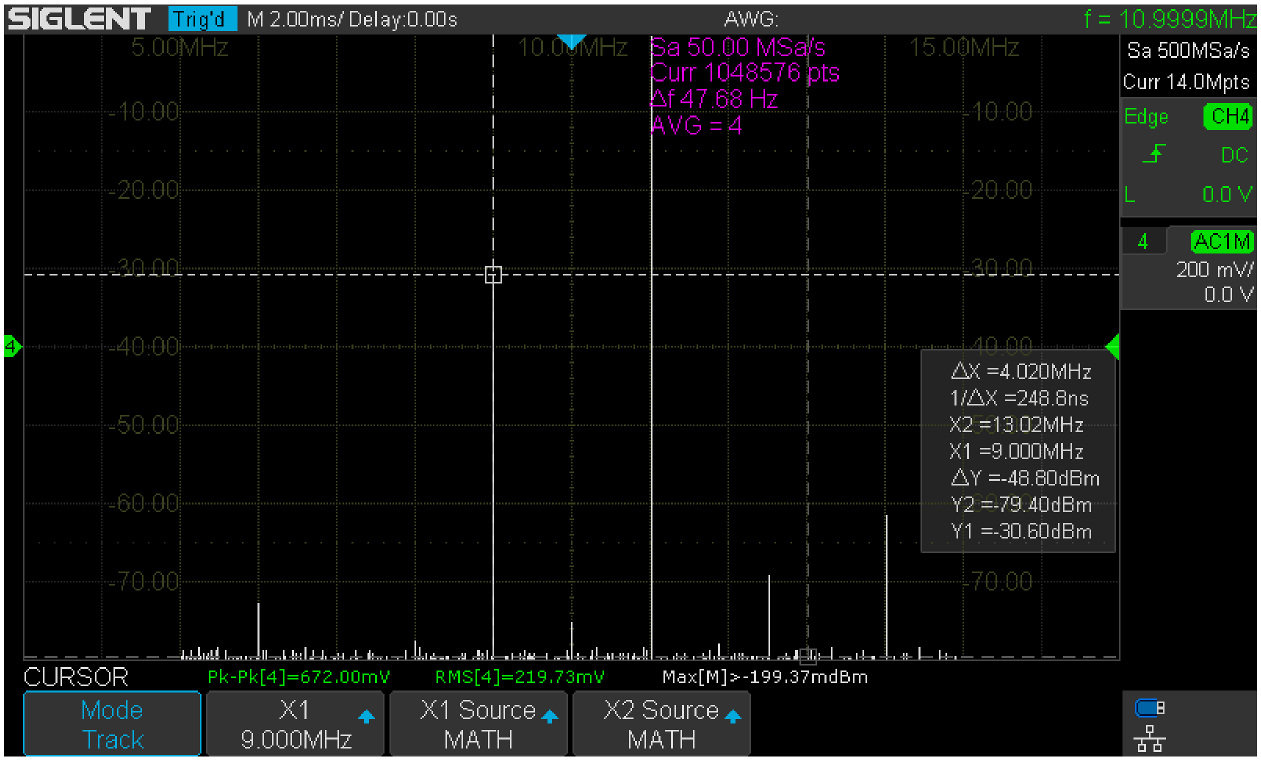


```
Sa 50.00 MSa/s
Curr 1048576 pts
Δf 47.68 Hz
AVG = 4

ΔX =4.020MHz
1/ΔX =248.8ns
X2 =13.02MHz
X1 =9.000MHz
ΔY =-51.60dBm
Y2 =-72.20dBm
Y1 =-20.60dBm

CURSOR  Pk-Pk[4]=704.00mV  RMS[4]=220.97mV  Max[M]>-189.39mdBm
```

SDS1104X-E_FFT_IM3_-20dBm

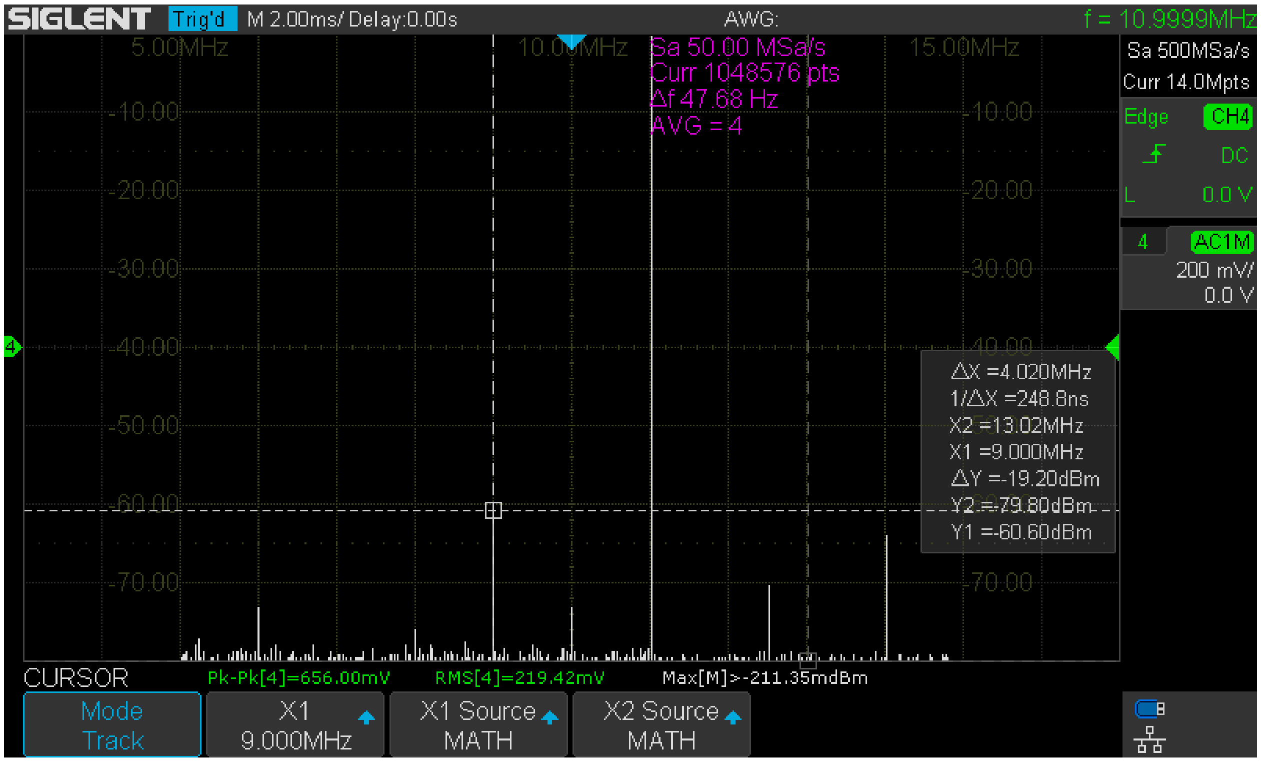


```
Sa 50.00 MSa/s
Curr 1048576 pts
Δf 47.68 Hz
AVG = 4

ΔX =4.020MHz
1/ΔX =248.8ns
X2 =13.02MHz
X1 =9.000MHz
ΔY =-48.80dBm
Y2 =-79.40dBm
Y1 =-30.60dBm

CURSOR  Pk-Pk[4]=672.00mV  RMS[4]=219.73mV  Max[M]>-199.37mdBm
```

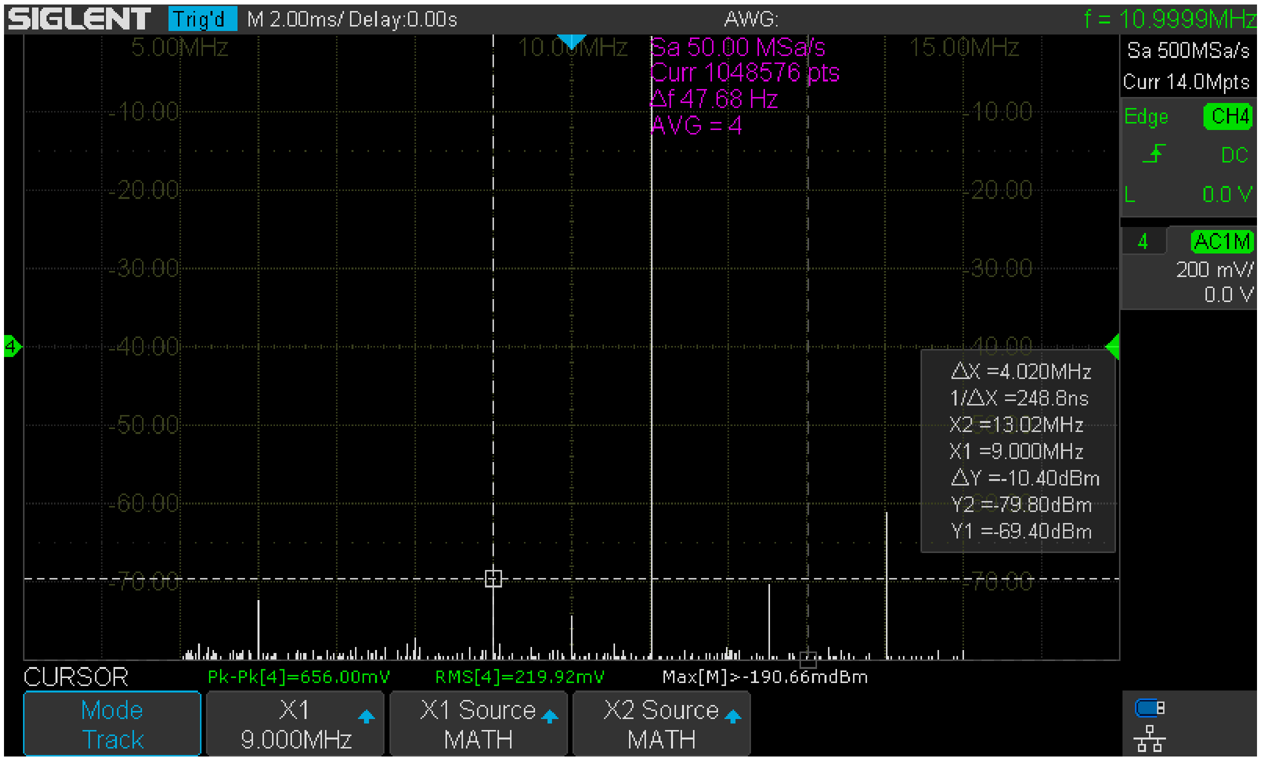
SDS1104X-E_FFT_IM3_-30dBm



ΔX =4.020MHz
1/ΔX =248.8ns
X2 =13.02MHz
X1 =9.000MHz
ΔY =-19.20dBm
Y2 =-79.80dBm
Y1 =-60.60dBm

Pk-Pk[4]=656.00mV RMS[4]=219.42mV Max[M]>-211.35mdBm

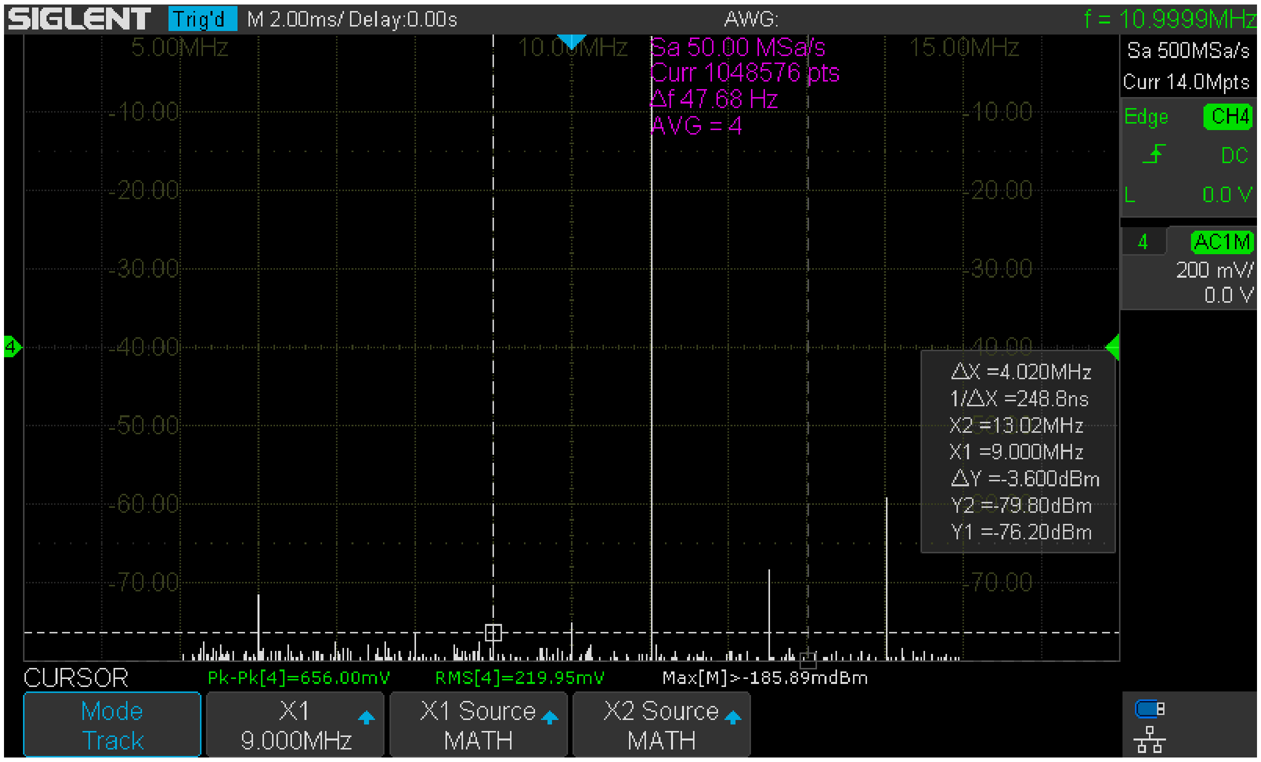
SDS1104X-E_FFT_IM3_-60dBm



ΔX =4.020MHz
1/ΔX =248.8ns
X2 =13.02MHz
X1 =9.000MHz
ΔY =-10.40dBm
Y2 =-79.80dBm
Y1 =-69.40dBm

Pk-Pk[4]=656.00mV RMS[4]=219.92mV Max[M]>-190.66mdBm

SDS1104X-E_FFT_IM3_-70dBm



SDS1104X-E_FFT_IM3_-76dBm

Application Examples

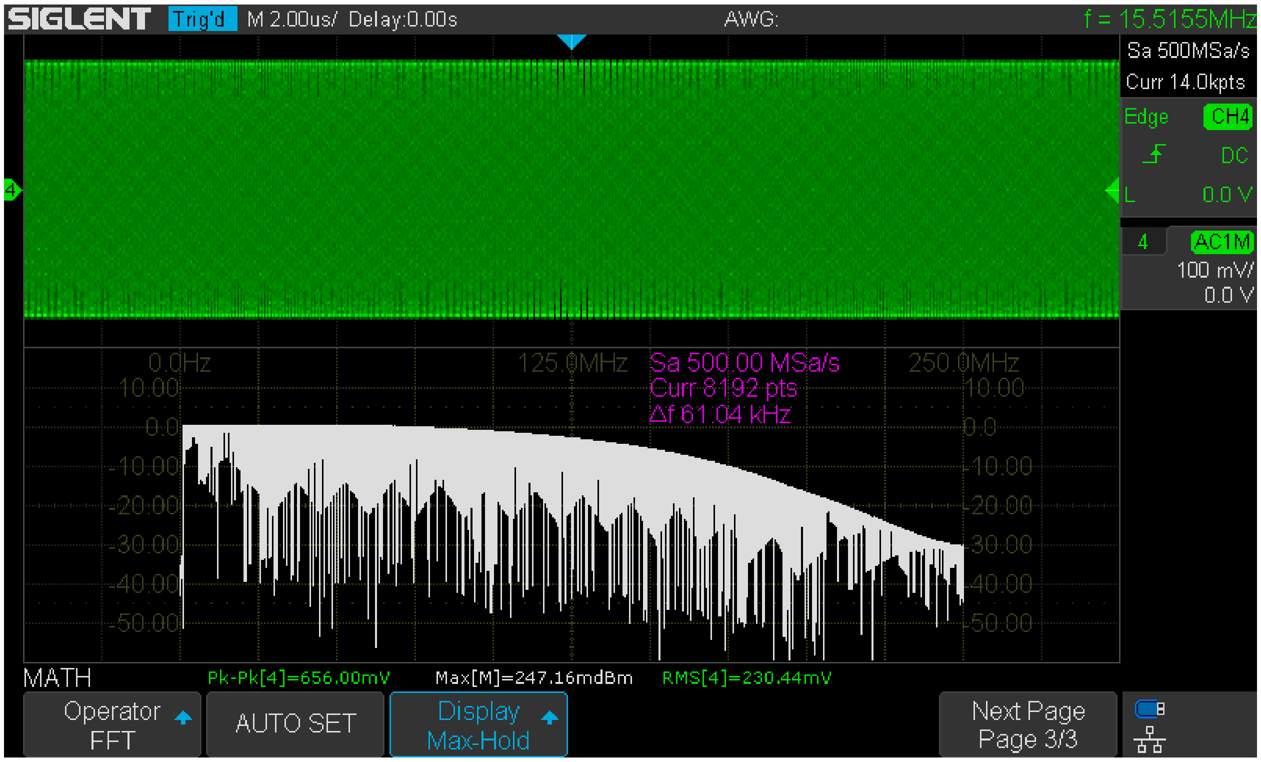
In this chapter we're looking at some more practical examples for using the SDS1104X-E FFT.

Wideband Measurement

We can plot the frequency response of the SDS1104X-E in the range of 1MHz to 250MHz using its FFT and an external signal source that covers the entire frequency range. There are three possibilities:

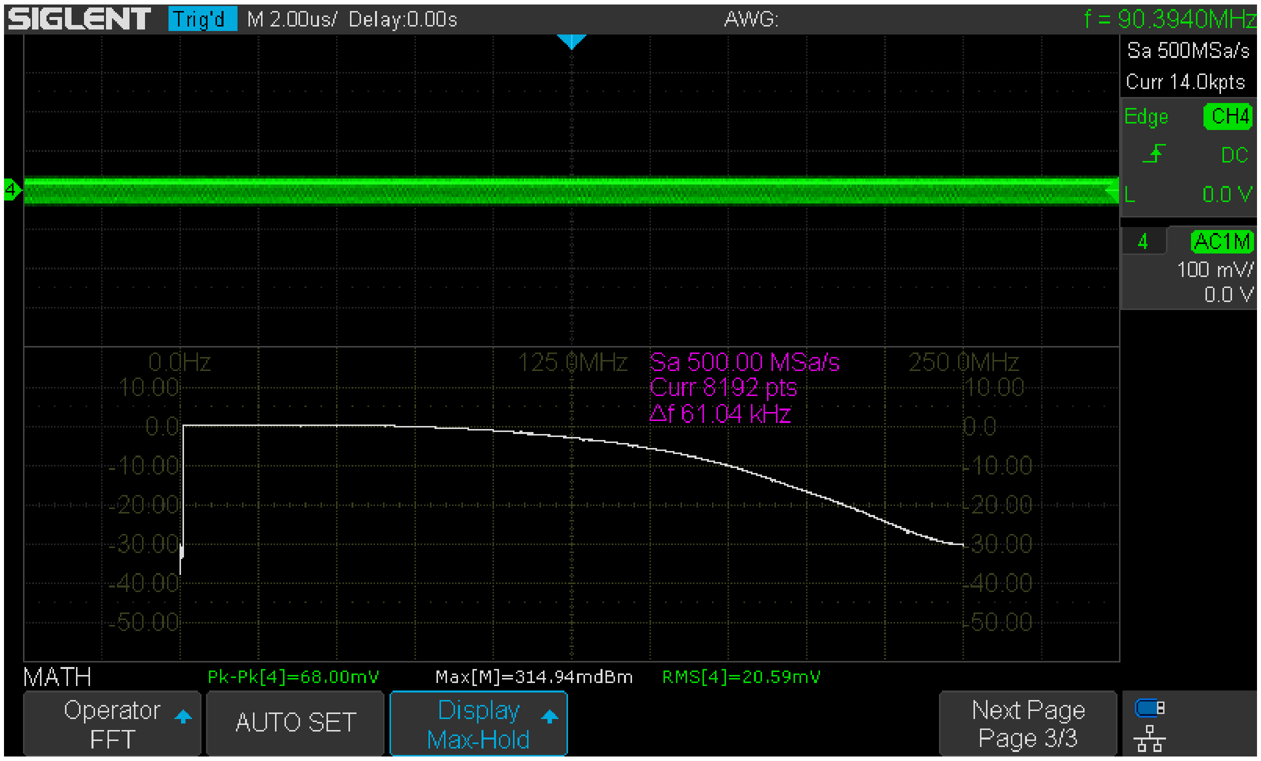
1. A wideband white noise source would ideally output a continuous spectrum of all frequencies at the same time. Unfortunately, such noise sources with a completely flat spectrum up to 250MHz aren't that easy to find and if so, they are certainly anything but cheap.
2. A pulse generator that outputs extremely narrow pulses with near zero transition time will create a discrete spectrum with a spectral line at constant amplitude for every *f* x *n*, where *f* is the repetition frequency of the pulse and *n* is any integer value from 1 to infinity. Once again, such a pulse generator is anything but common or cheap – e.g. 3.5ns pulse width and 1ns rise time would only be good up to some 30MHz.
3. A swept sine signal can provide near ideal amplitude accuracy, it just takes quite a while to build up the complete frequency response plot.

I've tried all three methods (using my limited resources), and unsurprisingly the third one yielded the best results by far. A sine wave with 0dBm amplitude, swept from 1MHz to 250MHz within 60 seconds was fed into channel 4 of the SDS1104X-E. The FFT has been setup for 250MHz analysis bandwidth, i.e. 2µs/div timebase and memory depth limited to 14kpts, yielding a frequency step of 61kHz, which is just perfect for this task.



Pk-Pk[4]=656.00mV Max[M]=247.16mdBm RMS[4]=230.44mV

SDS1104X-E_FFT_Sweep_1M-250MHz_init



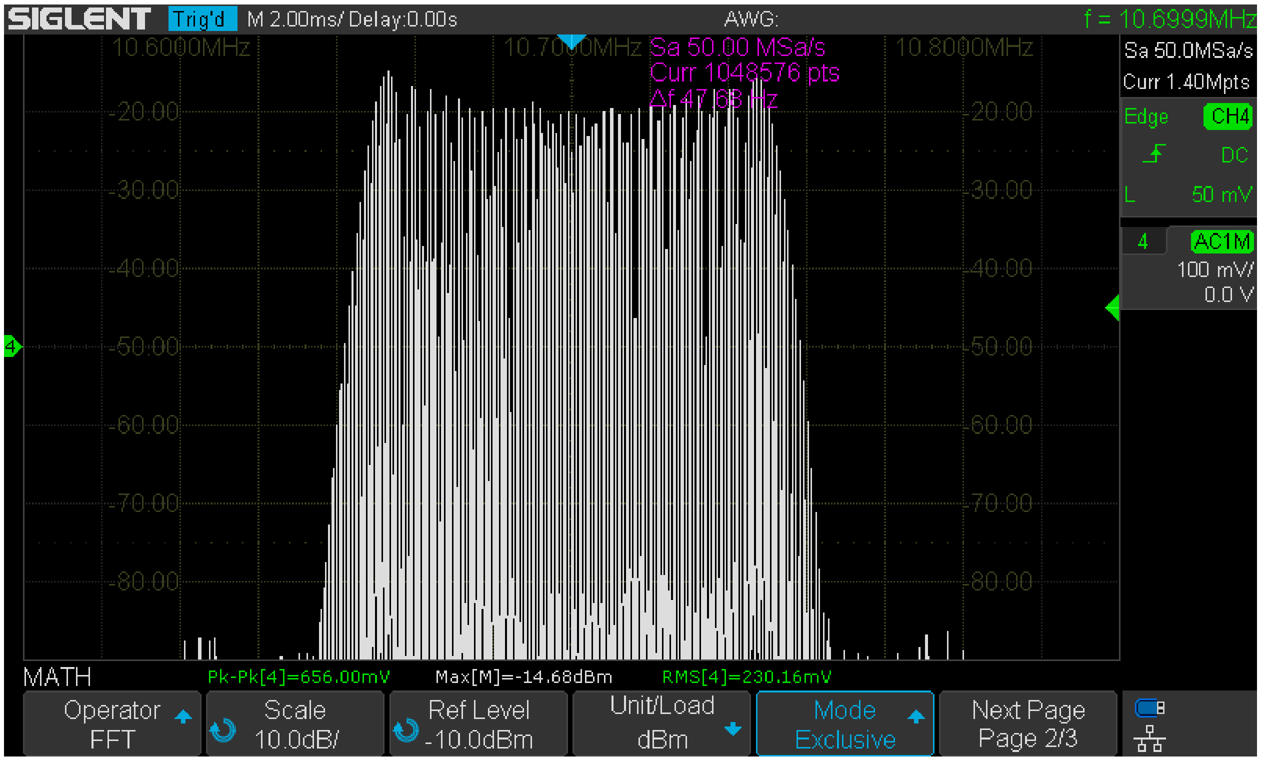
Pk-Pk[4]=68.00mV Max[M]=314.94mdBm RMS[4]=20.59mV

SDS1104X-E_FFT_Sweep_1M-250MHz_final

In order to collect all individual measurements, we use the Max-Hold Display mode. The first screenshot shows the trace after the first scan (after some 60 seconds), whereas the 2nd screenshot demonstrates how a near perfect frequency response graph can be obtained by just waiting several minutes until the maxima for all horizontal pixels have been registered. Of course, the same result could be obtained after the initial sweep if it were made much slower, but with the faster sweep we get a quick overview and then only need to wait any further if the result actually meets our expectations.

Narrowband Measurement

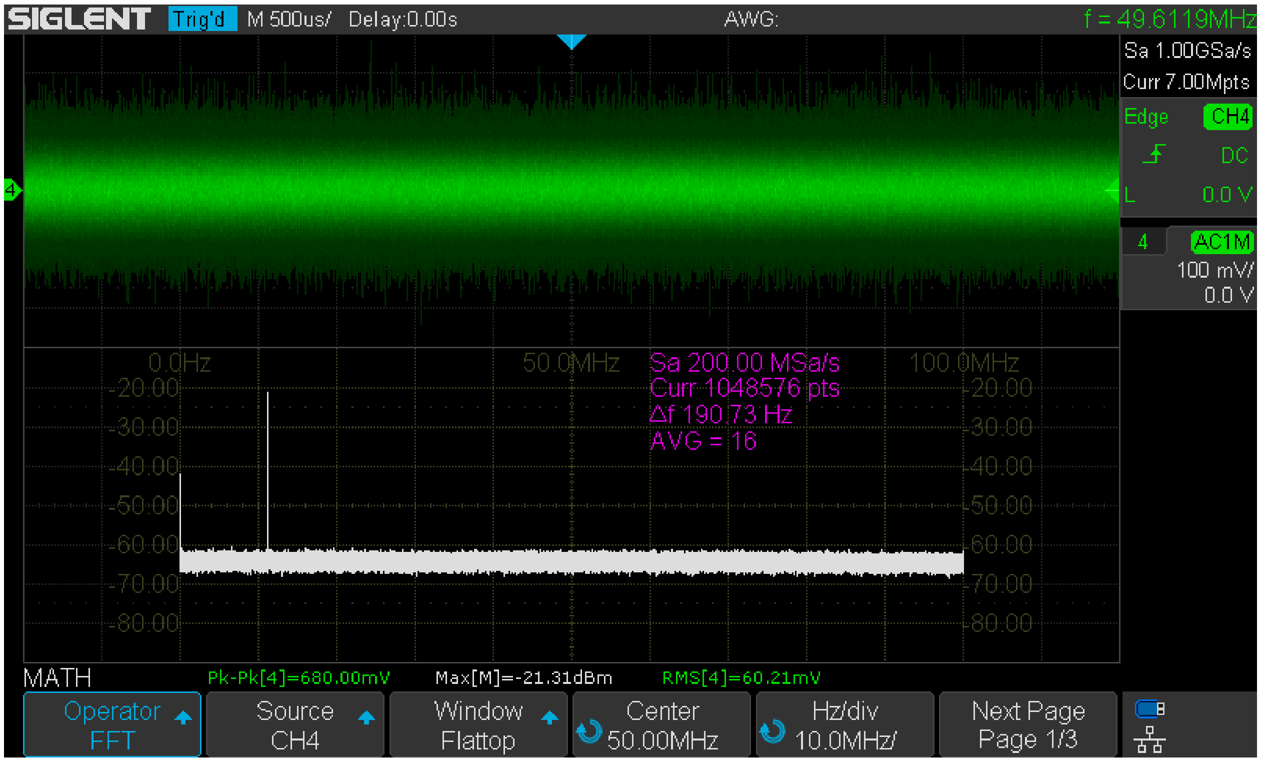
We can look at the IF signal of a domestic FM receiver at 10.7MHz, with a 1kHz frequency modulation and 50kHz max. deviation. For this we choose an analysis bandwidth of 25MHz – 12.5MHz would be even better, but we'd need to enable the 2nd channel in the group (Ch. 3 in this example) since this BW is only available in dual channel mode. We want the best possible frequency resolution, so we choose the longest FFT with 1048576 points, providing a frequency step of 47.7Hz. This leads us to 2ms/div timebase and 1.4Mpts memory depth and we just need to adjust the center frequency to 10.7MHz and set a span of 200kHz by selecting a horizontal scale of 20kHz/div.



SDS1104X-E_FFT_FM_10.7MHz

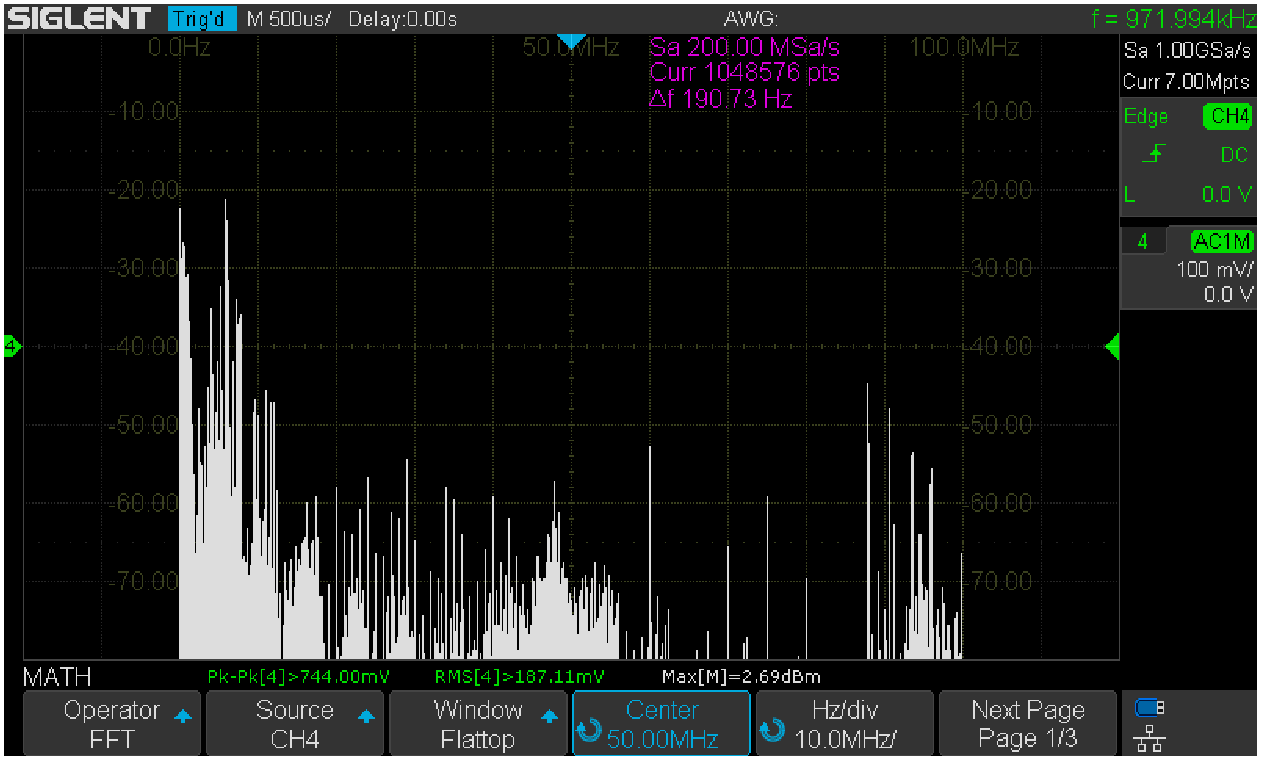
Signal burried in Noise

FFT can help to find weak signals completely buried in noise, hence invisible in the time domain. The example below shows a -20dBm 11.111MHz signal hidden under a 680mVpp noise floor. In the time domain, we can only see the noise and wouldn't even know about the -20dBm signal. The FFT over the full scope bandwidth shows that signal very clearly.



SDS1104X-E_Noise100mV_Sig-20dBm_11111kHz

Broadcast



SDS1104X-E_FFT_Broadcast

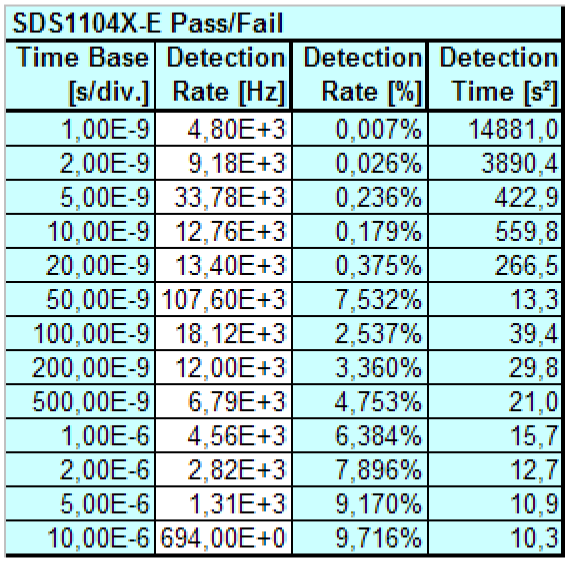
The Screenshot above shows the broadcast signals over a 100MHz bandwidth captured with just a piece of wire (about 10m long) as antenna. We can see everything from VLF up to the local VHF radio stations.

Mask Testing

This is usually not a very popular feature – and for a reason; it often comes as an expensive option and has the reputation to be a tool exclusively for production tests. The latter might also explain why its implementation is agonizingly slow on many scopes, just capable of analyzing a couple of acquisitions per second. This might be sufficient for production tests, but makes it almost useless for anything else.

Thankfully, Siglent went a different route as they have understood the potential of a more ambitious approach on mask test (Siglent calls it “Pass/Fail”) and it comes as a standard feature for free. The implementation on the X-series DSOs works pretty much at the same speed as normal acquisition and the maximum fault detection rate is close to the usual trigger rate aka “waveform update speed”. Consequently, mask testing can be a very useful tool for glitch hunting and fault finding, yielding a reasonably high probability for capturing a rare glitch. With one of the more common ridiculously slow implementations, it might take forever until a rare and very brief mask violation finally gets detected.

The table below shows the maximum fault detection rate for a 2MHz input signal at various timebase settings from 1ns/div up to 10µs/div. The table also shows the expected detection rate in percent of the total number of faults as well as the average time for detecting a glitch that has a repetition rate of 1Hz.



SDS1104X-E Pass/Fail			
Time Base [s/div.]	Detection Rate [Hz]	Detection Rate [%]	Detection Time [s²]
1,00E-9	4,80E+3	0,007%	14881,0
2,00E-9	9,18E+3	0,026%	3890,4
5,00E-9	33,78E+3	0,236%	422,9
10,00E-9	12,76E+3	0,179%	559,8
20,00E-9	13,40E+3	0,375%	266,5
50,00E-9	107,60E+3	7,532%	13,3
100,00E-9	18,12E+3	2,537%	39,4
200,00E-9	12,00E+3	3,360%	29,8
500,00E-9	6,79E+3	4,753%	21,0
1,00E-6	4,56E+3	6,384%	15,7
2,00E-6	2,82E+3	7,896%	12,7
5,00E-6	1,31E+3	9,170%	10,9
10,00E-6	694,00E+0	9,716%	10,3

SDS1104X-E_Mask_Test_Detection_Rate

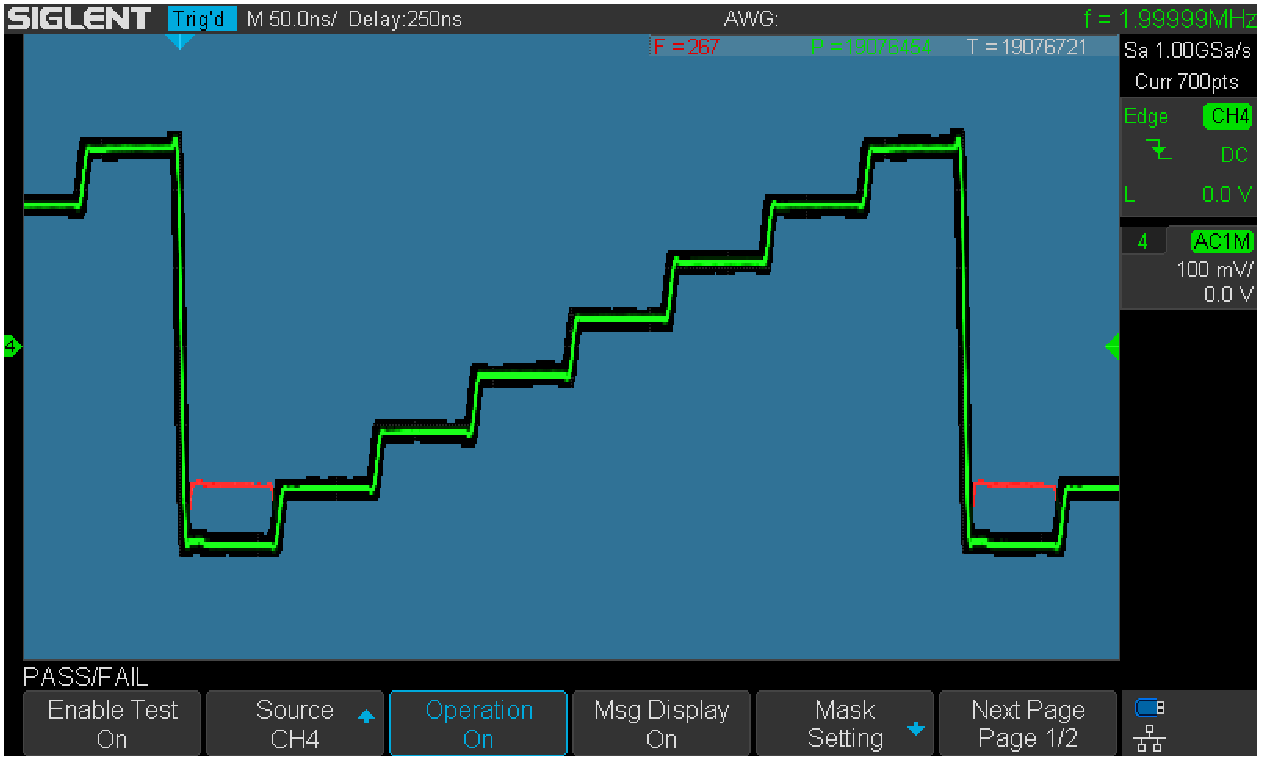
As an example, let’s assume a step-up curve where the bottom step is occasionally missing. The signal repetition rate is 2MHz and the fault occurs at 20Hz, hence an original fault rate of 1:100000 or 0.001%.

At the sweet spot of 50ns/div we get a max. fault detection rate of 107600 acquisitions per second, hence a theoretical probability for capturing the fault of

Fault_detection_rate x *timebase* x *horizontal_divisions* = 107600 x 50e-9 x 14 = 0.07532 = 7.532%;

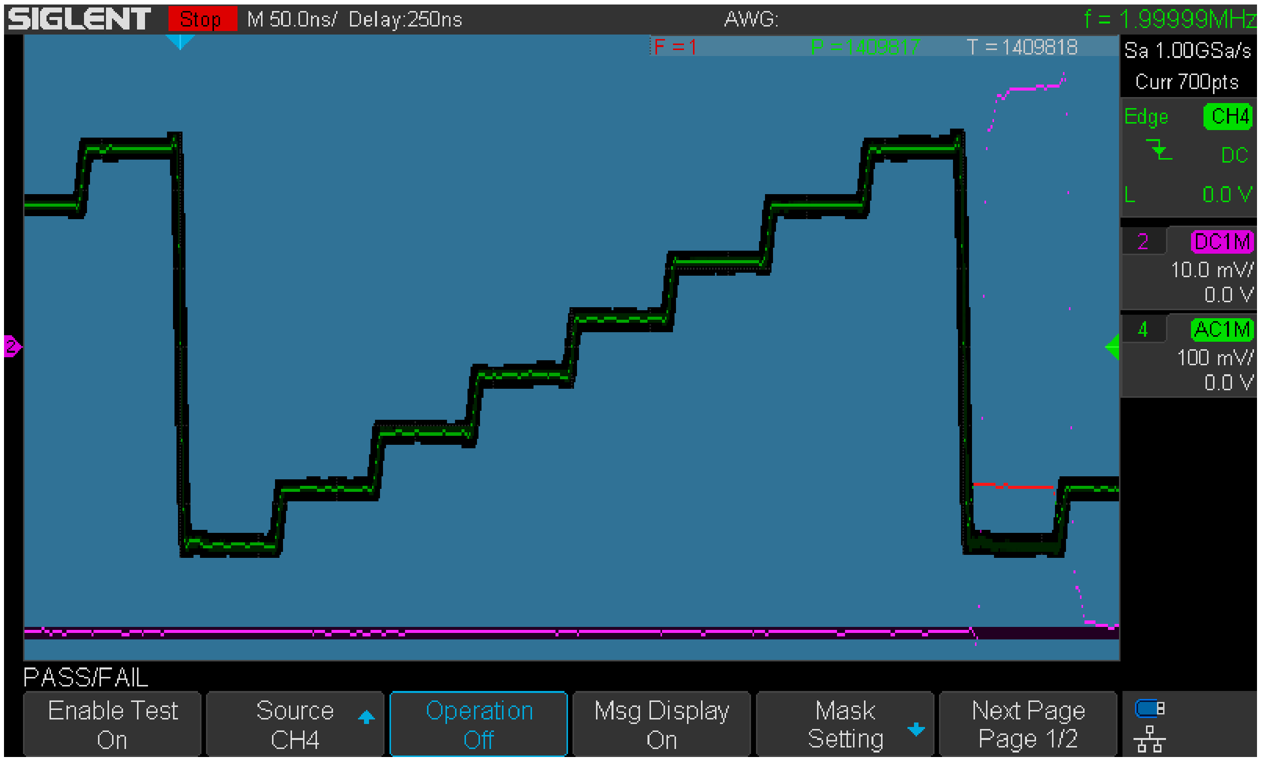
The screenshot below demonstrates this very situation, showing the pass/fail statistics after 177 seconds runtime: a total number of 19076721 (19 million!) tests and 267 fault detections. The actual fault rate was 20Hz, resulting in a total of 20 x 177 = 3540 faults where 267 got detected. 267 / 3540 = 0.0754 = 7.54%; Average detection time was 177s / 267 = 0.663s and for 1Hz fault rate this would be 0.663s x 20 = 13.26s. So these measurements confirm the theoretical numbers given in the table above very nicely.

Don’t forget to turn display persistence on in order to clearly see where the mask violations have occurred. It’s the red traces in the screenshot, indicating that the bottom step gets cropped occasionally. With this information, one could easily find an appropriate (e.g. runt) trigger to capture the glitch itself and look for related signals that might be causing the fault directly or indirectly.



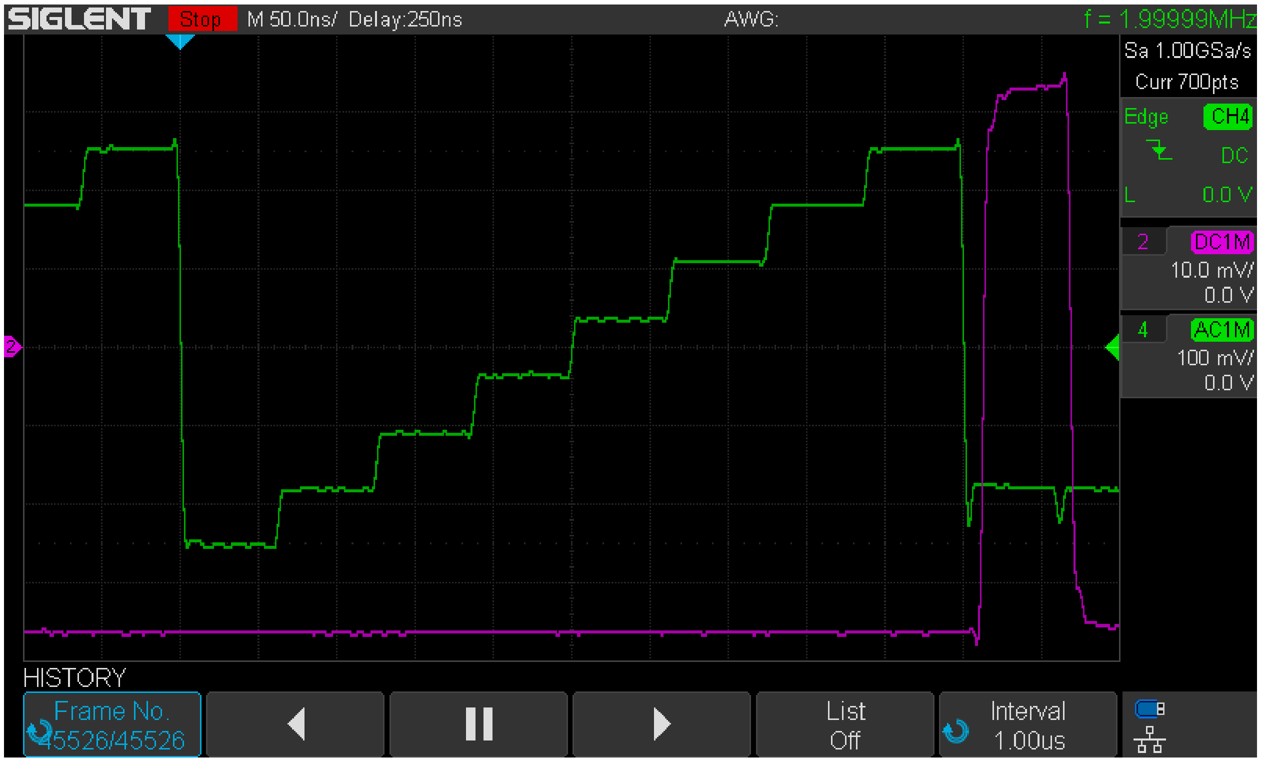
SDS1104X-E_Masktest_50ns_2MHz_20Hz_177s

Another option is enabling the *Stop on Fail* option on page 2 of the *PASS/FAIL* menu. This will stop the acquisition after the first mask violation has been detected, as shown in the screenshot below.

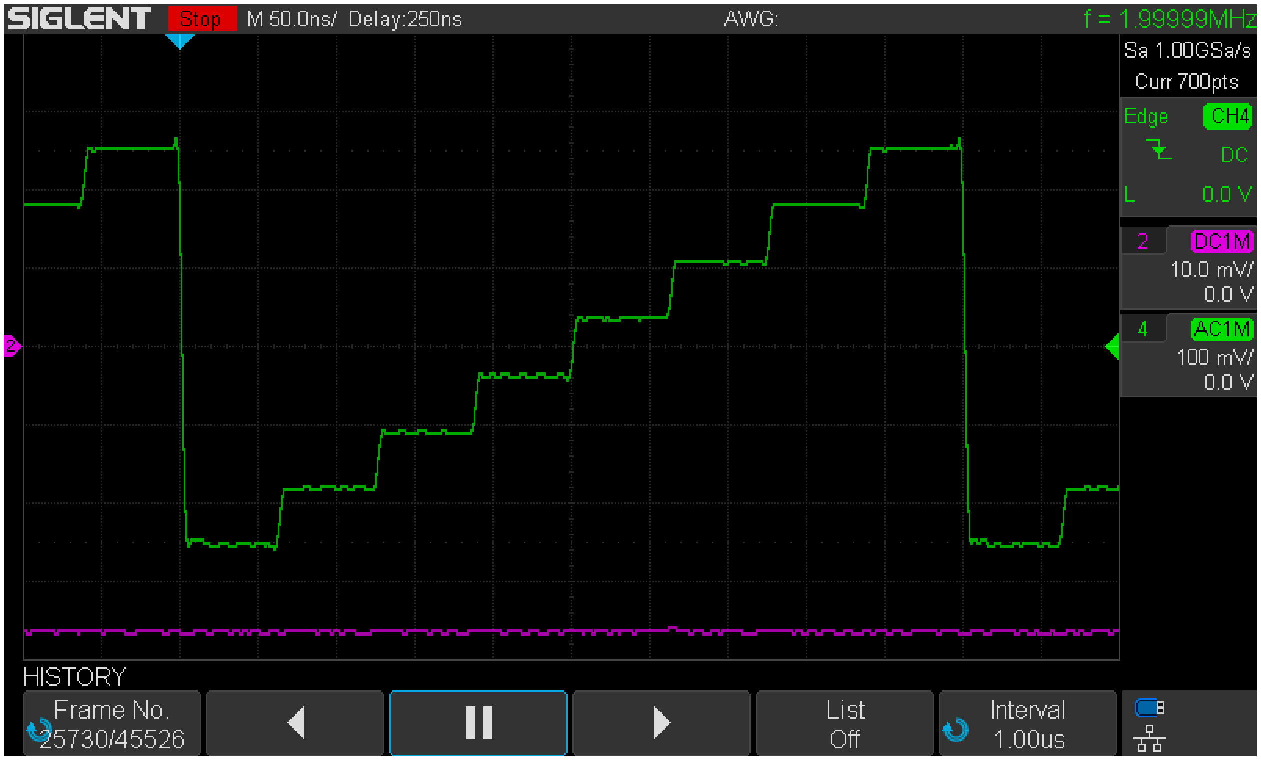


SDS1104X-E_Masktest_50ns_2MHz_20Hz_Stop

Another channel is used to monitor a signal that is suspected to be related to the fault and sure enough it is, as can be seen in this screenshot already thanks to the display persistence. Now we can disable mask test and enter the history:



SDS1104X-E_Masktest_50ns_2MHz_20Hz_Hist_Last



SDS1104X-E_Masktest_50ns_2MHz_20Hz_Hist

The first screenshot shows the last history frame that holds the acquisition with the mask violation. We can clearly see that the signal at channel 2 is closely related to this event. Any other history frame just shows the normal signal without a fault together with the inactive signal on channel 2 and the 2nd screenshot gives an example for this.

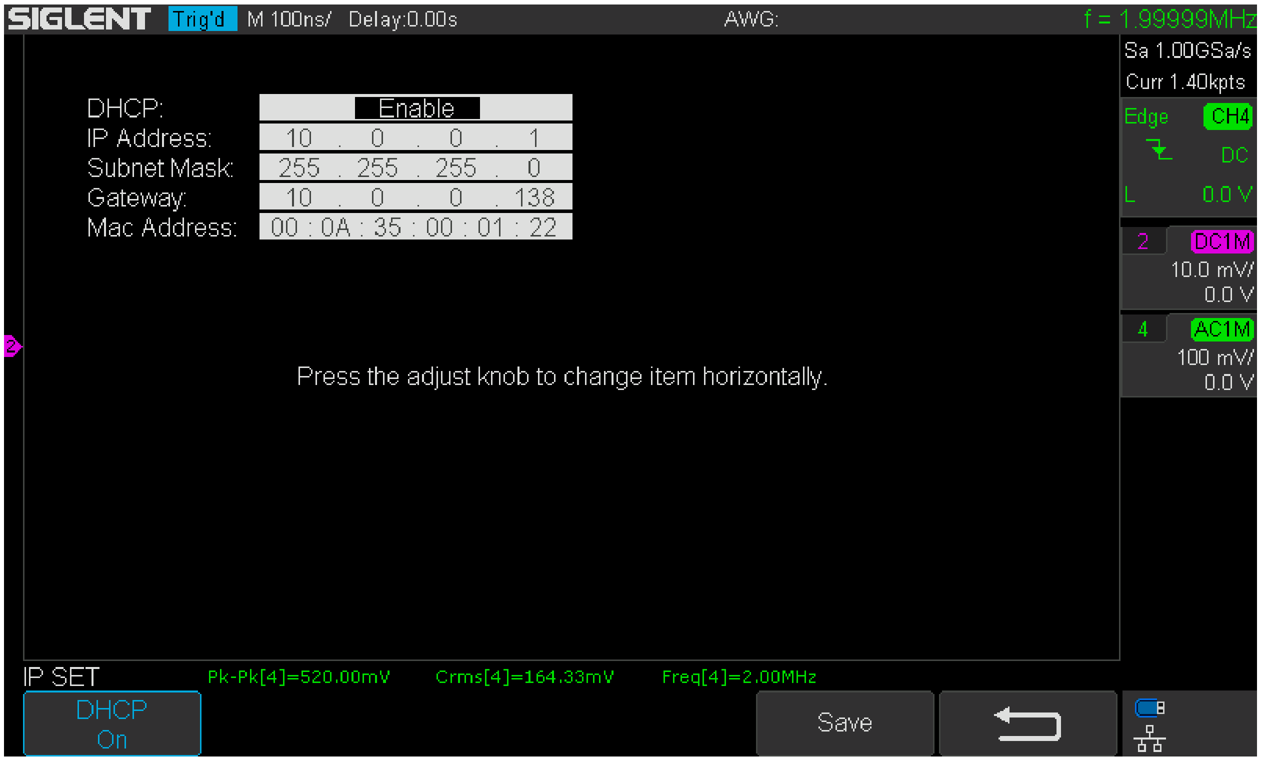
You might ask what the advantages of mask testing are compared to just using (infinite) display persistence alone. Well, there is a couple…

- We might have less stationary signals that require wider tolerances on the mask setting. With such an unstable signal, any violation would be harder to spot with just persistence alone, whereas in the mask test, they stick out in red color.
- Mask test provides some statistics if *Msg Display* is turned on. By knowing the number of tests and faults, we can estimate the fault rate, which might give valuable hints on where to look for the culprit.
- We can set mask test to stop after the first occurrence of a mask violation. This allows the close examination of the situation including related signals.

We can calculate the fault rate by dividing the detected faults by the total number of tests. In this example, it would be 267 / 19076721 ~ 14e-6; this is not exactly the true fault rate which would be 20Hz/2MHz = 10e-6, but certainly close enough to make a guess what other signal might be related to the glitch. We would only consider slow signals (or conditions) with a repetition rate <30Hz and could concentrate on observing these. Likewise, it could be a problem in our firmware and we would use a GPIO to generate a signal that allows us to monitor a critical region in the firmware with the scope and see whether the fault consistently occurs when the code in that region is being executed.

Web Server

For the first time, a Siglent DSO has a built-in web server. It is nothing exciting, and does not show a live view of the DSO screen, so it's by no means a substitute for an USB scope. It just allows remote setup of the most common functions and pulling individual screenshots. A nice first shot nevertheless…



DHCP:	Enable
IP Address:	10 . 0 . 0 . 1
Subnet Mask:	255 . 255 . 255 . 0
Gateway:	10 . 0 . 0 . 138
Mac Address:	00 : 0A : 35 : 00 : 01 : 22

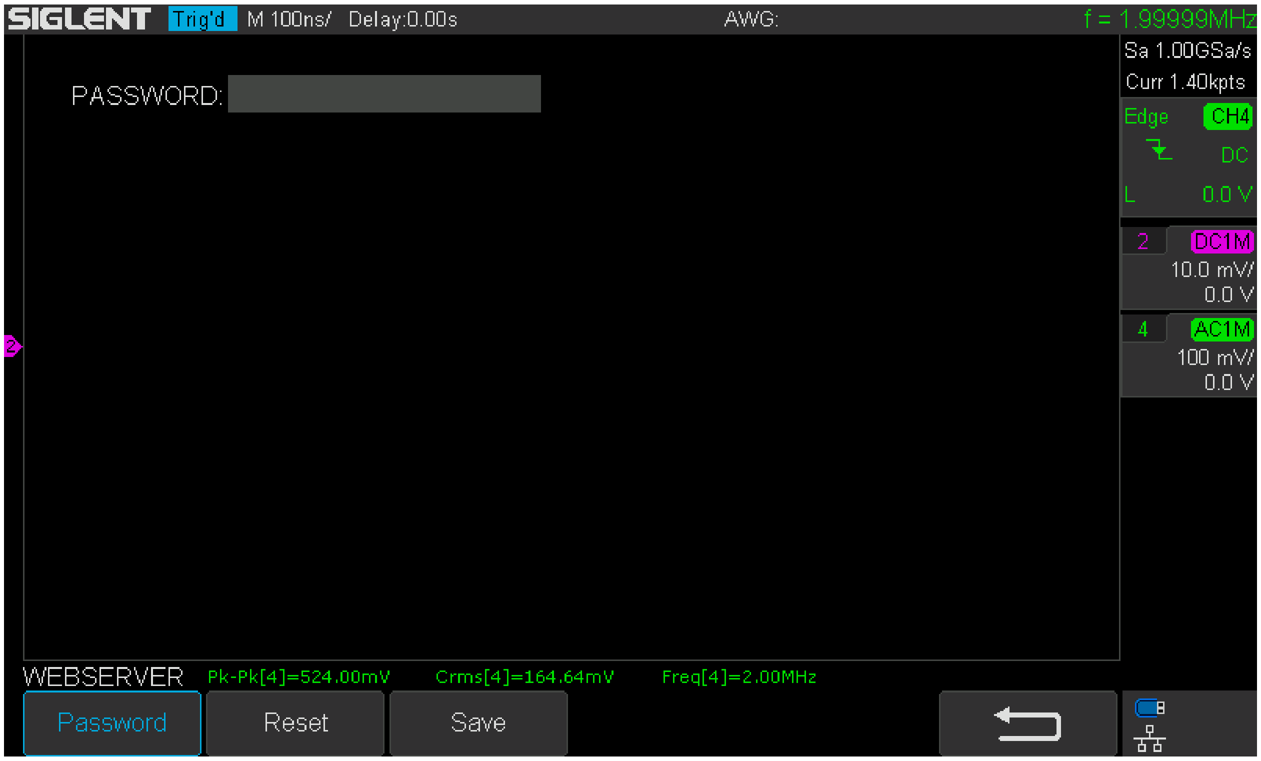
Press the adjust knob to change item horizontally.

Pk-Pk[4]=520.00mV Crms[4]=164.33mV Freq[4]=2.00MHz

SDS1104X-E_IP_setting

On page 2 of the *Utility* menu, we can set up the IP configuration. This is totally easy as long as a network router with integrated DHCP server is available. All that's needed is enabling DHCP and waiting a moment until the scope has received its individual IP address, see screenshot above. In a simpler network without DHCP server, the IP address, Subnet Mask and Gateway have to be set manually – I have tried that as well and it works as expected.

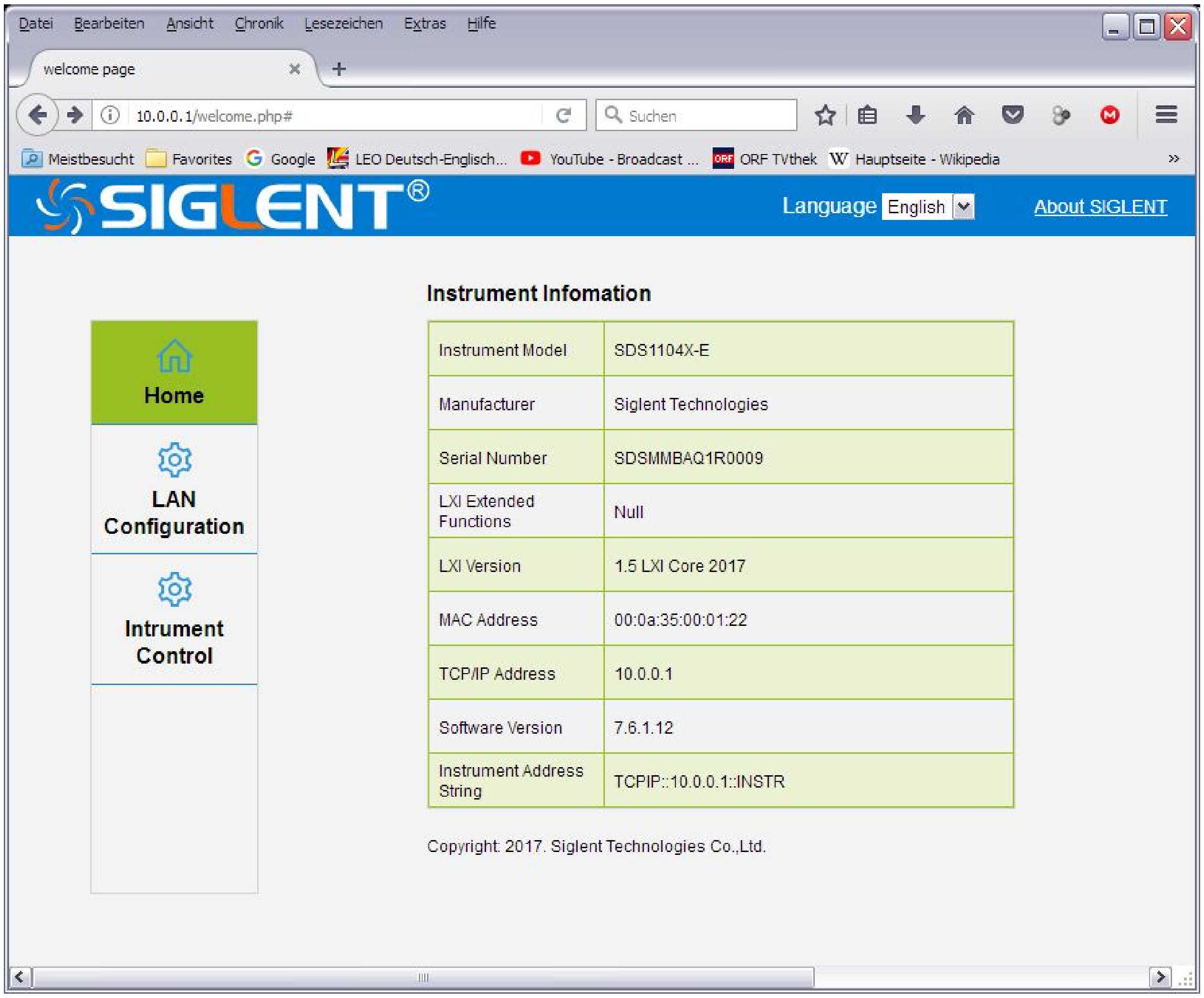
On page 4 of the *Utility* menu a password for web server access can be set. I haven't bothered to do this, as I hate passwords (which I tend to forget) and my local network isn't accessible from the internet.



Pk-Pk[4]=524.00mV Crms[4]=164.64mV Freq[4]=2.00MHz

SDS1104X-E_WEB_Password

On the local computer, we just need to open a web browser and type the scope's IP address into the address bar. After hitting [**Return**] the home screen of our web server appears.


Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
welcome page
10.0.0.1/welcome.php#
Suchen
Meistbesucht Favorites Google LEO Deutsch-Englisch... YouTube - Broadcast ... ORF TVthek Hauptseite - Wikipedia
SIGLENT®
Language English
About SIGLENT
Home
LAN Configuration
Intrument Control

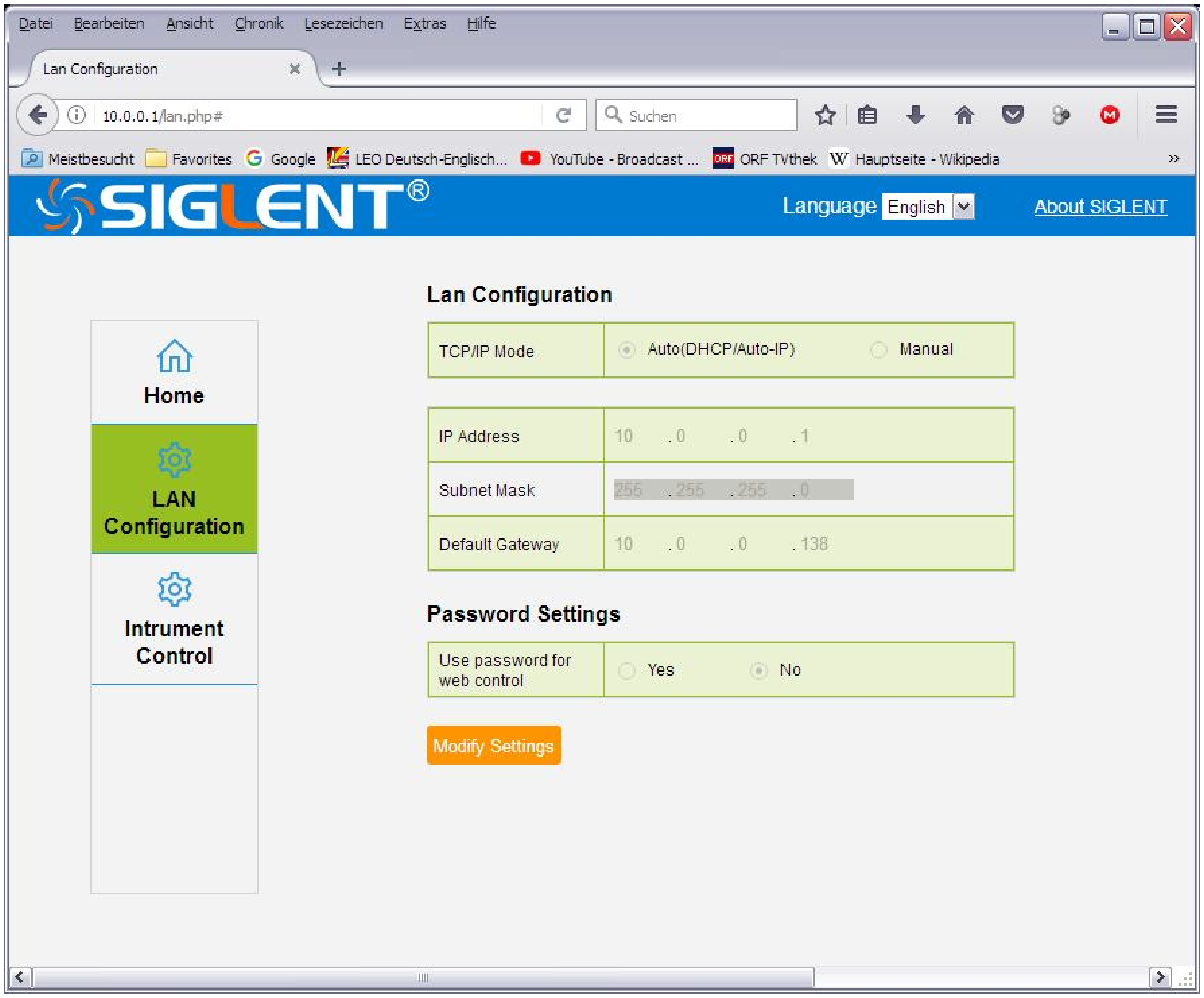
Instrument Infomation

Instrument Model	SDS1104X-E
Manufacturer	Siglent Technologies
Serial Number	SDSMMBAQ1R0009
LXI Extended Functions	Null
LXI Version	1.5 LXI Core 2017
MAC Address	00:0a:35:00:01:22
TCP/IP Address	10.0.0.1
Software Version	7.6.1.12
Instrument Address String	TCPIP::10.0.0.1::INSTR

Copyright: 2017. Siglent Technologies Co.,Ltd.

WEB_Start

This contains some useful information about the instrument, like model, serial number and firmware version. This screen offers access to the LAN configuration as well as the Instrument Control. Let's have a look at LAN Configuration first.


Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Lan Configuration
10.0.0.1/lan.php#
Suchen
Meistbesucht Favorites Google LEO Deutsch-Englisch... YouTube - Broadcast ... ORF TVthek Hauptseite - Wikipedia
SIGLENT®
Language English
About SIGLENT
Home
LAN Configuration
Intrument Control
Modify Settings

Lan Configuration

TCP/IP Mode	[✓ Auto(DHCP/Auto-IP)] Manual
IP Address	10 . 0 . 0 . 1
Subnet Mask	255 . 255 . 255 . 0
Default Gateway	10 . 0 . 0 . 138

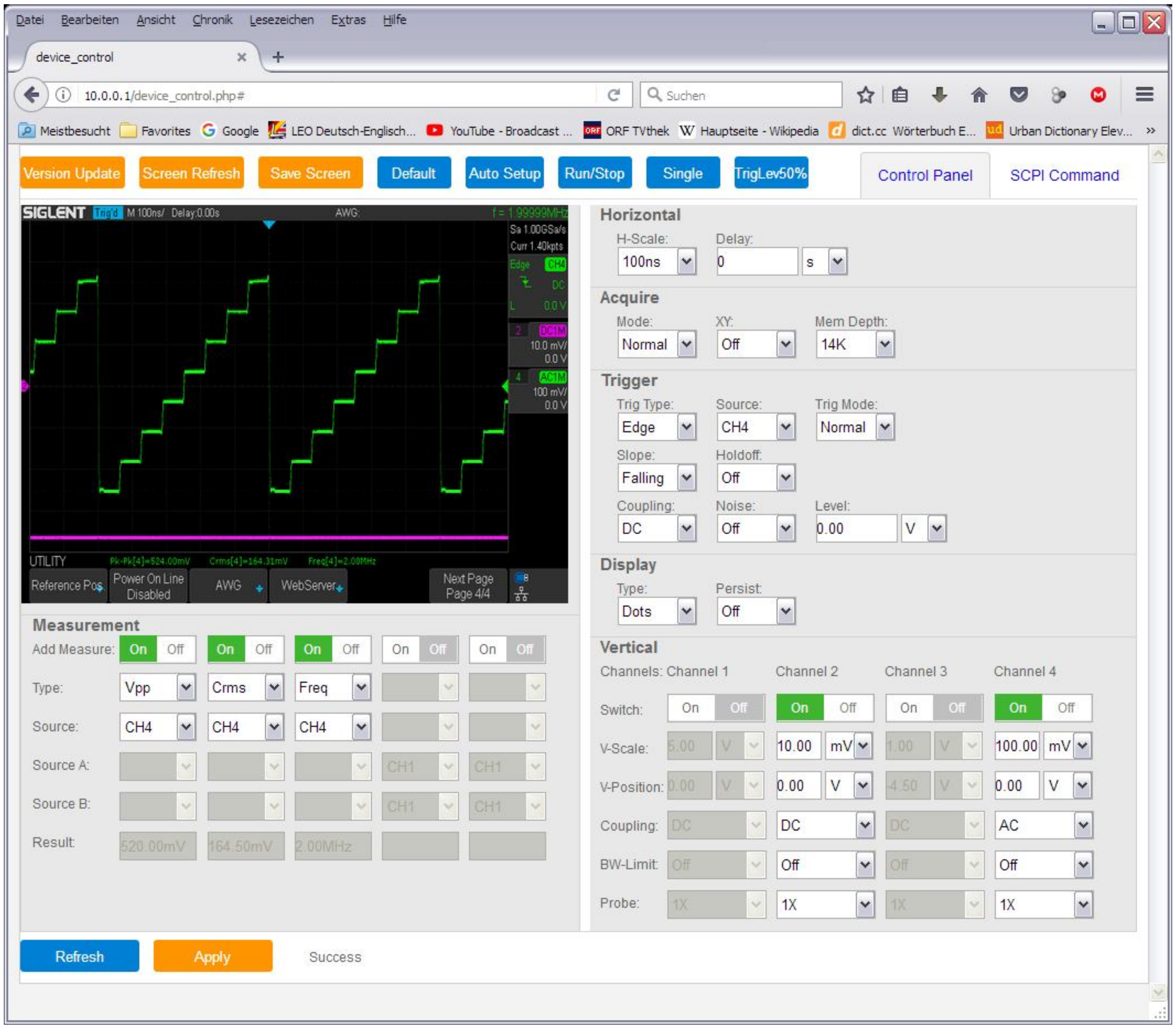
Password Settings

Use password for web control	Yes [✓ No]

WEB_Settings

Nothing fancy here and I assume the screenshot is self-explanatory.

So let’s move on to the actually interesting part, the Instrument Control.



Horizontal

H-Scale: 100ns Delay: 0 s

Acquire

Mode: Normal XY: Off Mem Depth: 14K

Trigger

Trig Type: Edge Source: CH4 Trig Mode: Normal

Slope: Falling Holdoff: Off

Coupling: DC Noise: Off Level: 0.00 V

Display

Type: Dots Persist: Off

Vertical

Channels:	Channel 1	Channel 2	Channel 3	Channel 4
Switch:	Off	On	Off	On
V-Scale:	5.00 V	10.00 mV	1.00 V	100.00 mV
V-Position:	0.00 V	0.00 V	-4.50 V	0.00 V
Coupling:	DC	DC	DC	AC
BW-Limit:	Off	Off	Off	Off
Probe:	1X	1X	1X	1X

Measurement

Add Measure:	On	On	On	Off	Off
Type:	Vpp	Crms	Freq		
Source:	CH4	CH4	CH4		
Source A:				CH1	CH1
Source B:				CH1	CH1
Result:	520.00mV	164.50mV	2.00MHz		

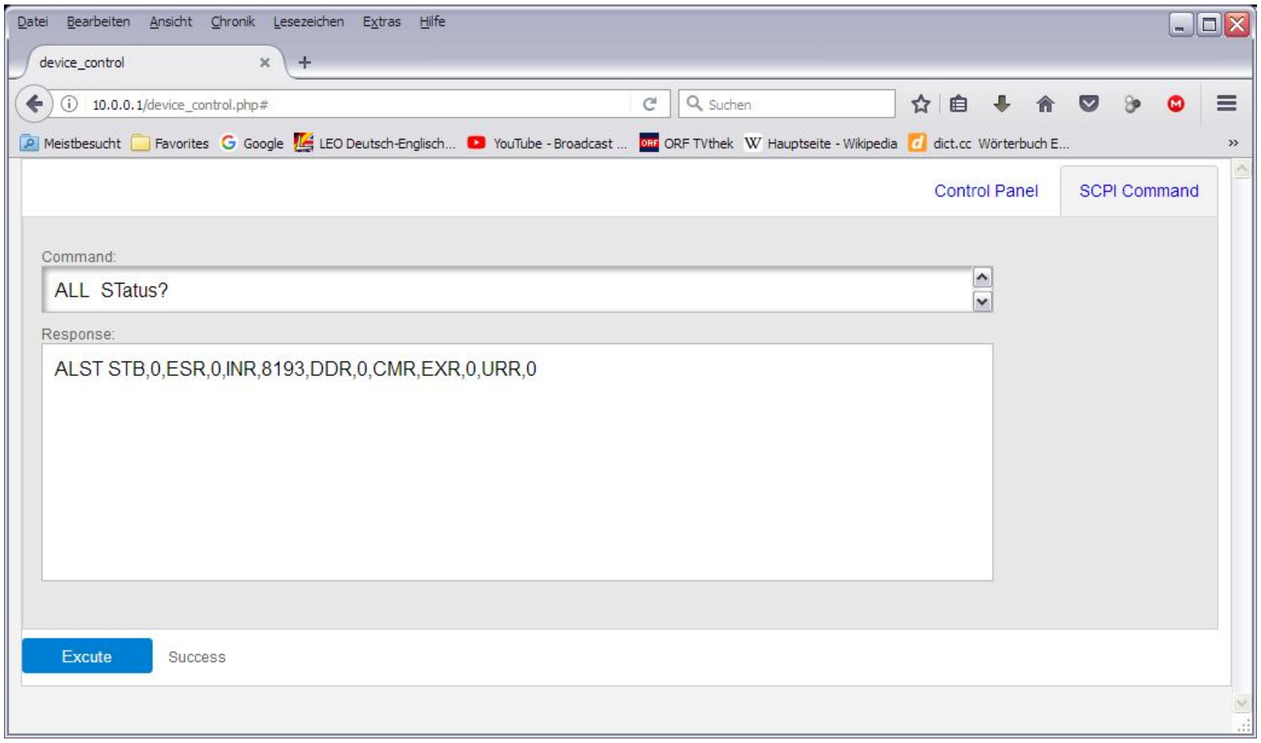
WEB_Control

We get an initial snapshot of the DSO screen and can do a screen refresh anytime we want – it just isn’t fast; even on a Gbit network, it takes up to 3s to load a new screenshot, at least on my ancient computer.

We can configure the most basic settings, but even for this, not all options are available. For example, we cannot choose the interpolation method in the Acquire section and the Display section does not even offer Color mode. Likewise, not all trigger types are available. The Vertical section does not provide the Deskew and Invert Options. The available Measurements appear to be fairly complete, but here again, the options for Gate, Statistics and All Measurements are missing. All in all it’s a really basic control application that doesn’t support any of the more advanced features of the SDS1104X-E. I’m not sure if it really would be sensible to offer much more, as a true remote operation of the scope makes little sense without a live view anyway. I look at the web server more as a demonstration tool to play with remote commands and the most common tasks would be getting screenshots from the scope without the need for an USB flash-drive.

Screenshots can be saved by clicking *Save Screen* and then a bitmap file is stored in the web browser’s default download directory. Alternatively, the usual methods for getting an image from a website can be applied, like right-click on the screen image and select “Save graphics as…” from the context menu. In any case, a bitmap file is stored, whereas we want it to be PNG. My preferred method is right-click, select “Copy Graphics” from the context menu, then paste it to an image processing program – I prefer Microsoft Office Picture Manager for simple tasks like this – and then export it as .png file from there.

The Instrument Control page offers yet another nice option; this is the “SCPI Command” tab in the upper right corner. With this we get a terminal window and can talk to the instrument via SCPI without the need for running a terminal program on the computer. Of course this is absolutely basic and mainly serves for playing around with the scope and trying out various commands. The example below shows the response to an ALL_STatus? command.



Command:

```
ALL_STatus?
```

Response:

```
ALST STB,0,ESR,0,INR,8193,DDR,0,CMR,EXR,0,URR,0
```

WEB_SCPI_ALL_ST