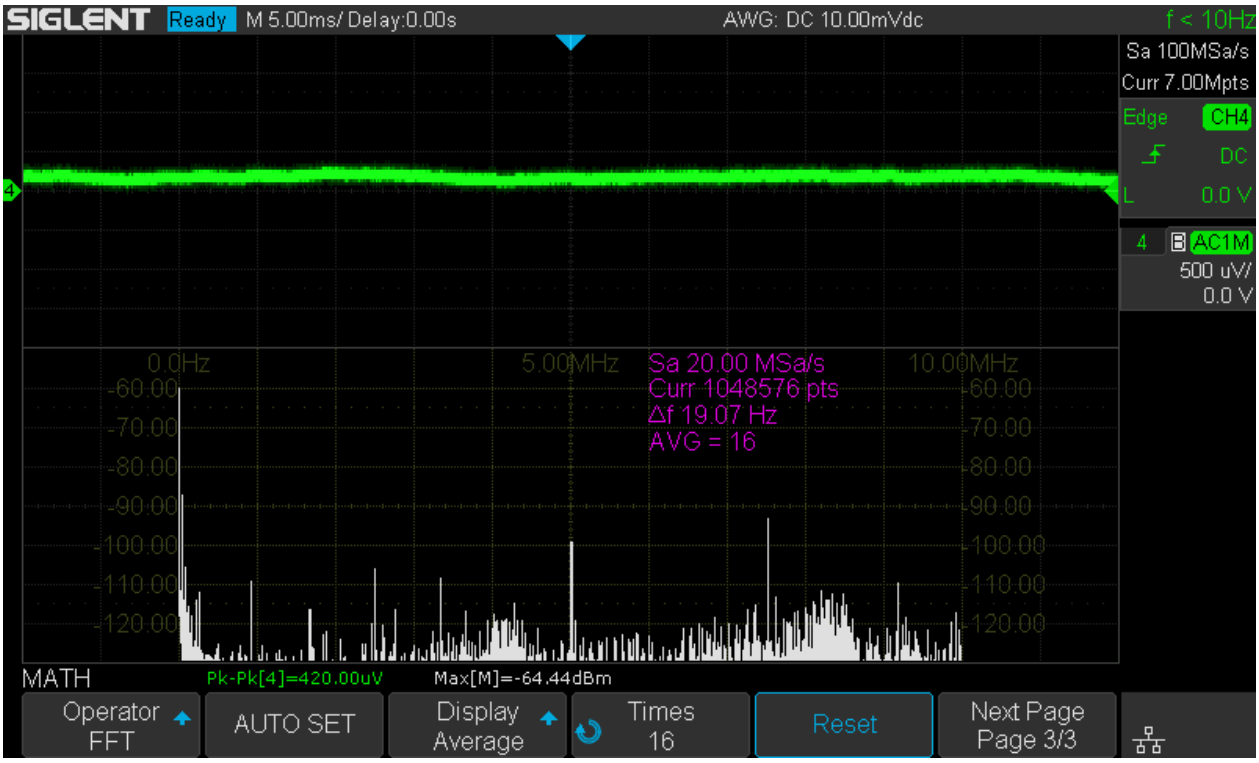
Siglent SAG1021 Review

The DC “signal” shouldn’t actually be a signal, but just a plain DC voltage without any AC components. To check this, I took the opportunity to use the excellent FFT of the SDS1104X-E in order to detect any interference signals. The full DC output voltage was connected to the AC-coupled DSO input channel 4 and the spectrum from 0 to 10MHz computed:



SAG1021_DC_10mV_FFT

As can be seen from the screenshot above, even at maximum sensitivity of 500µV/div, the Y-t trace shows nothing but white noise. Even the FFT with a 19Hz frequency step could not detect any signal stronger than -92dBm (5.62μV_{RMS}), which could just as well be an interference from the scope rather than the function generator.

Arbitrary Waveforms

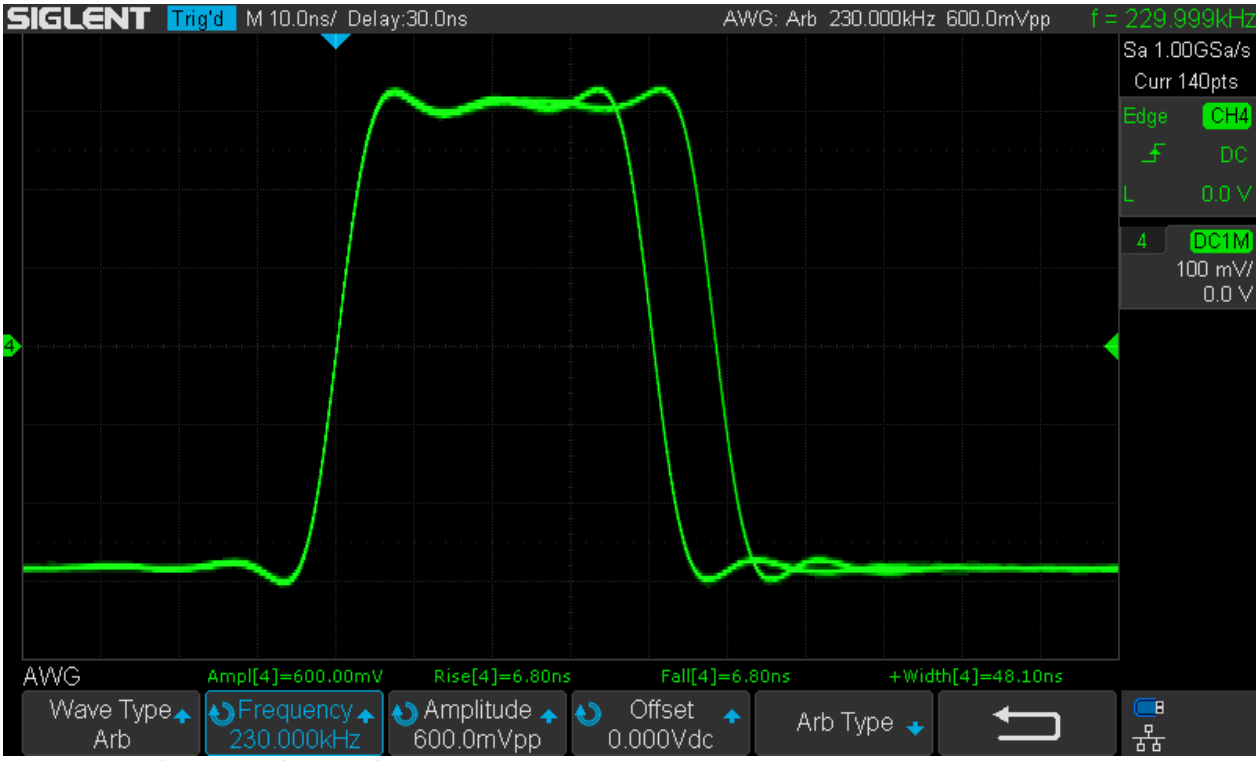
The SAG1021 can handle arbitrary waveforms up to 16kpts at a sample rate of 125MSa/s and a maximum repetition rate up to 5MHz. A set of predefined arbitrary waveforms is stored internally and user defined waveforms are supported as well.

Internal

There are 45 predefined waveforms presented in a single dialog box as shown in the screenshot below.

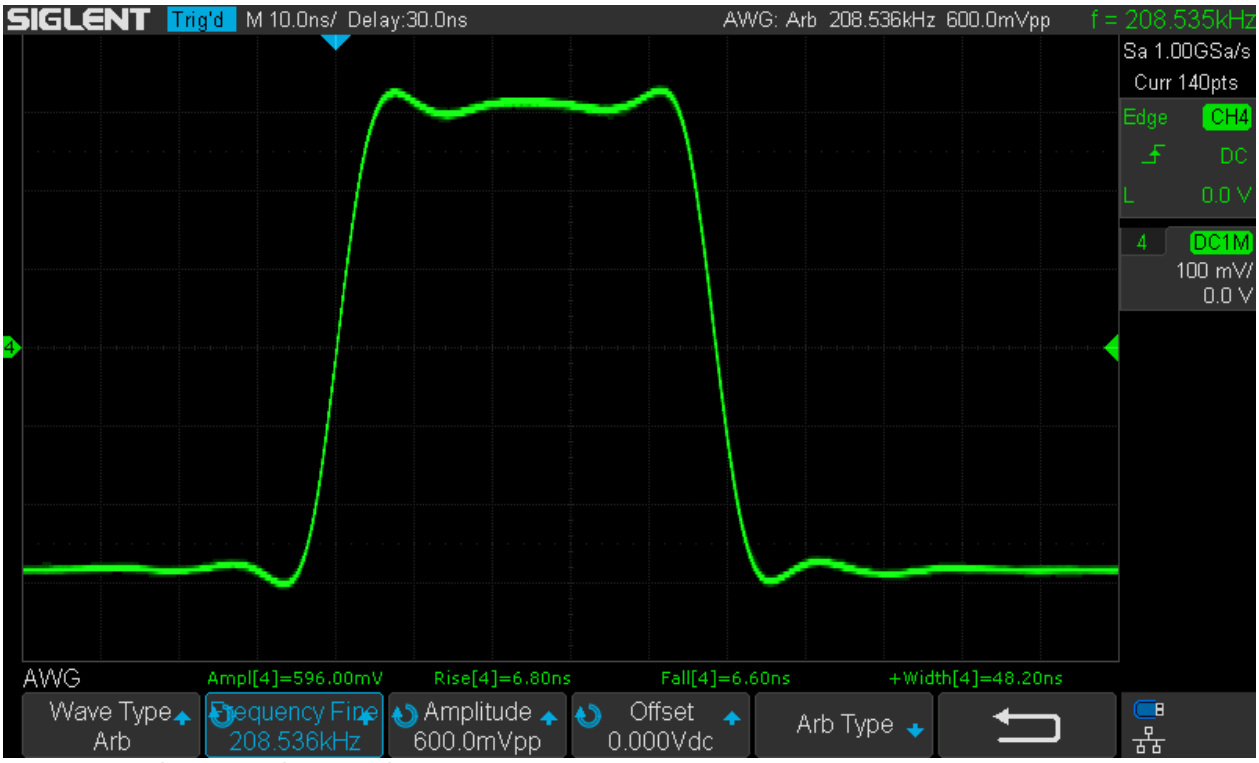
SAG1021_Arb_Int

One interesting waveform is the positive pulse *Ppulse*. It is of course not like the regular pulse function; as an arbitrary waveform it can only be a square wave with a certain duty cycle, which is 1% here. At random frequencies there is quite some jitter, as is typical for DDS generated waveforms:

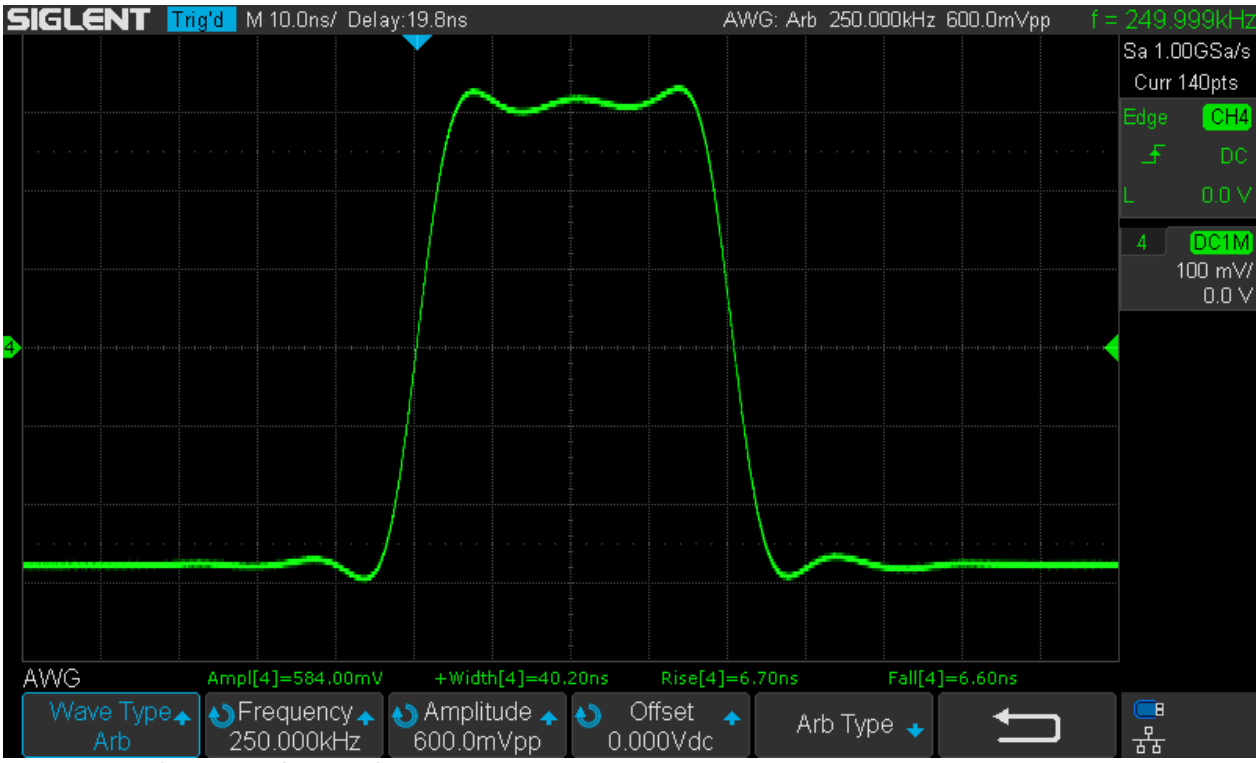


SAG1021_Arb_Int_Ppulse_230kHz

Only certain frequencies allow consistent sampling of the arbitrary waveform and yield a jitter-free output. 208.536kHz is such a frequency and its actual value depends on several parameters, like sample clock and number of samples used for that particular arbitrary waveform. Below we can see jitter-free outputs at both the tweaked 208.536kHz and a nice round 250kHz:

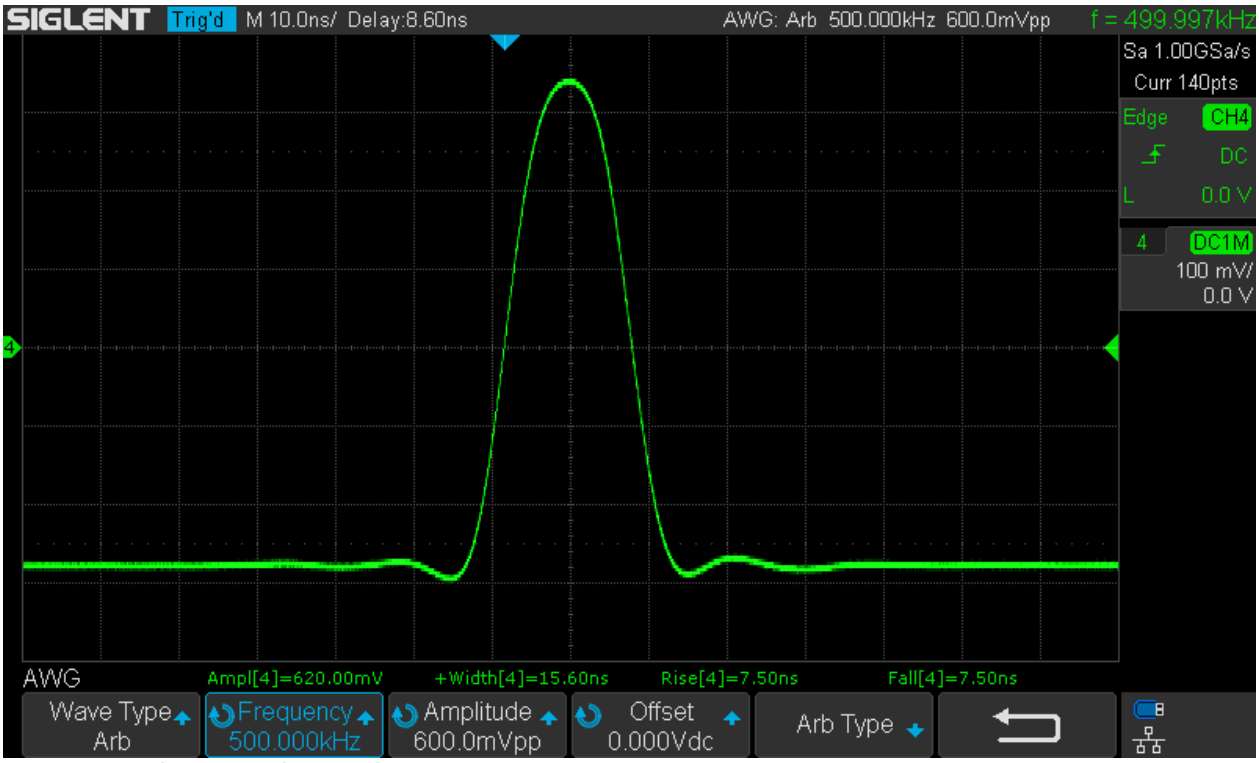


SAG1021_Arb_Int_Ppulse_tweak

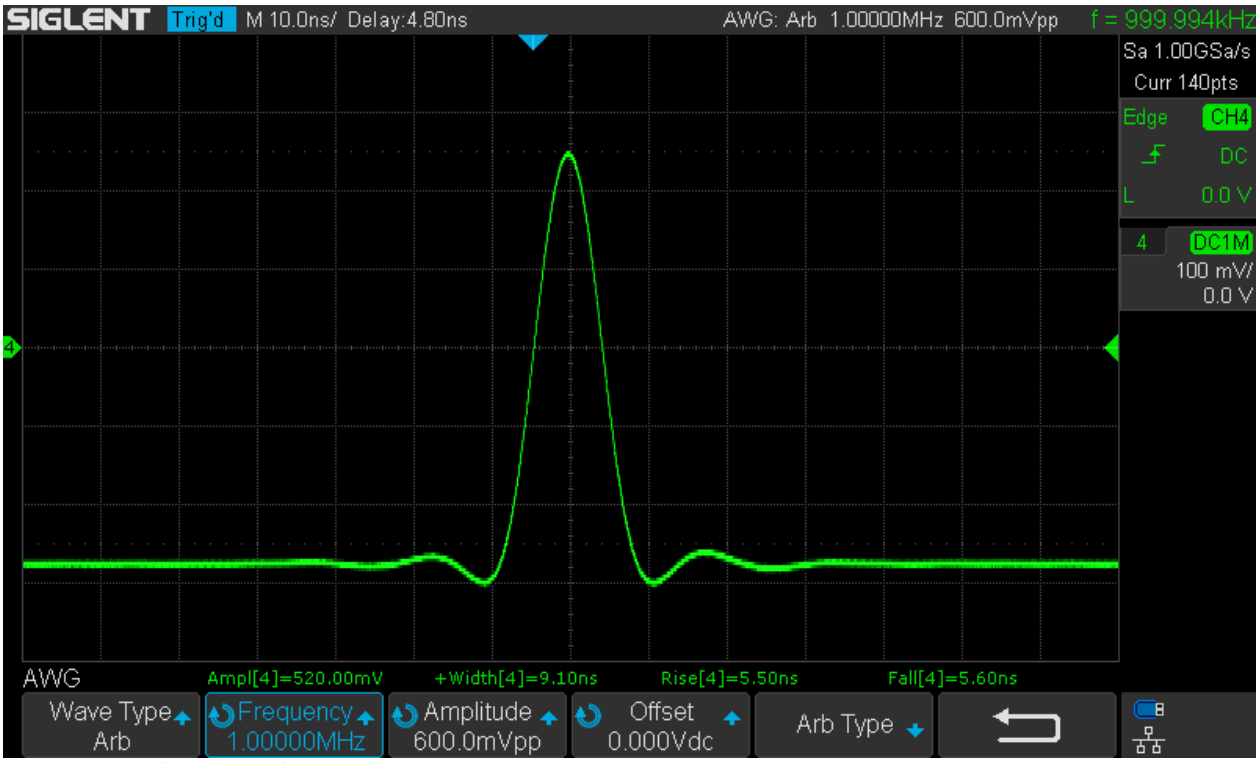


SAG1021_Arb_Int_Ppulse_250kHz

At 250kHz we already got a pulse width shorter than what the standard pulse function provides. This is only possible, because the transition times are much faster as well at <7.5ns. At 500kHz, we still get full amplitude with a pulse width of just 15.6ns and at 1MHz we finally get <10ns, with reduced output amplitude but even faster transition times around 5.6ns:

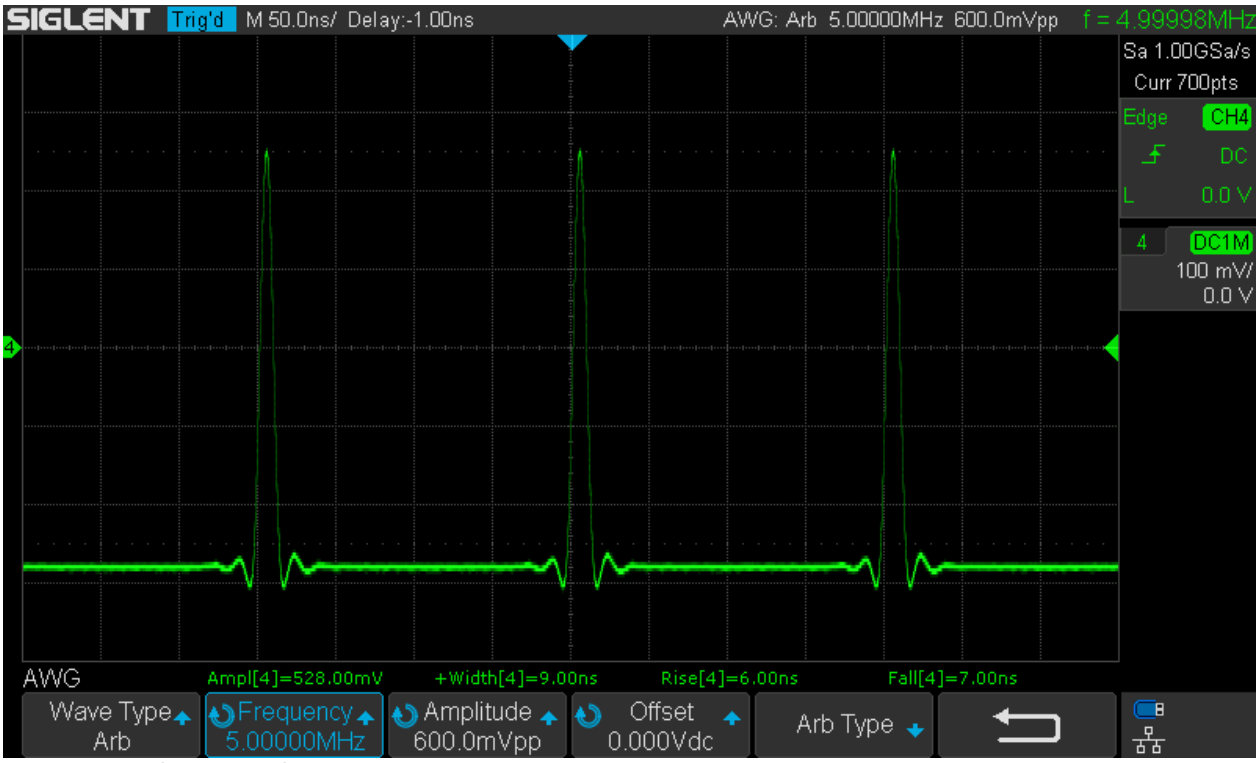


SAG1021_Arb_Int_Ppulse_500kHz



SAG1021_Arb_Int_Ppulse_1MHz

At even higher frequencies, the pulse properties don't change any further and we get pretty much the same pulse at a higher repetition rate. The transition times haven't changed either, just the measurements are less accurate by one nanosecond at slower timebases and the measurements keep flickering between 6 and 7ns.



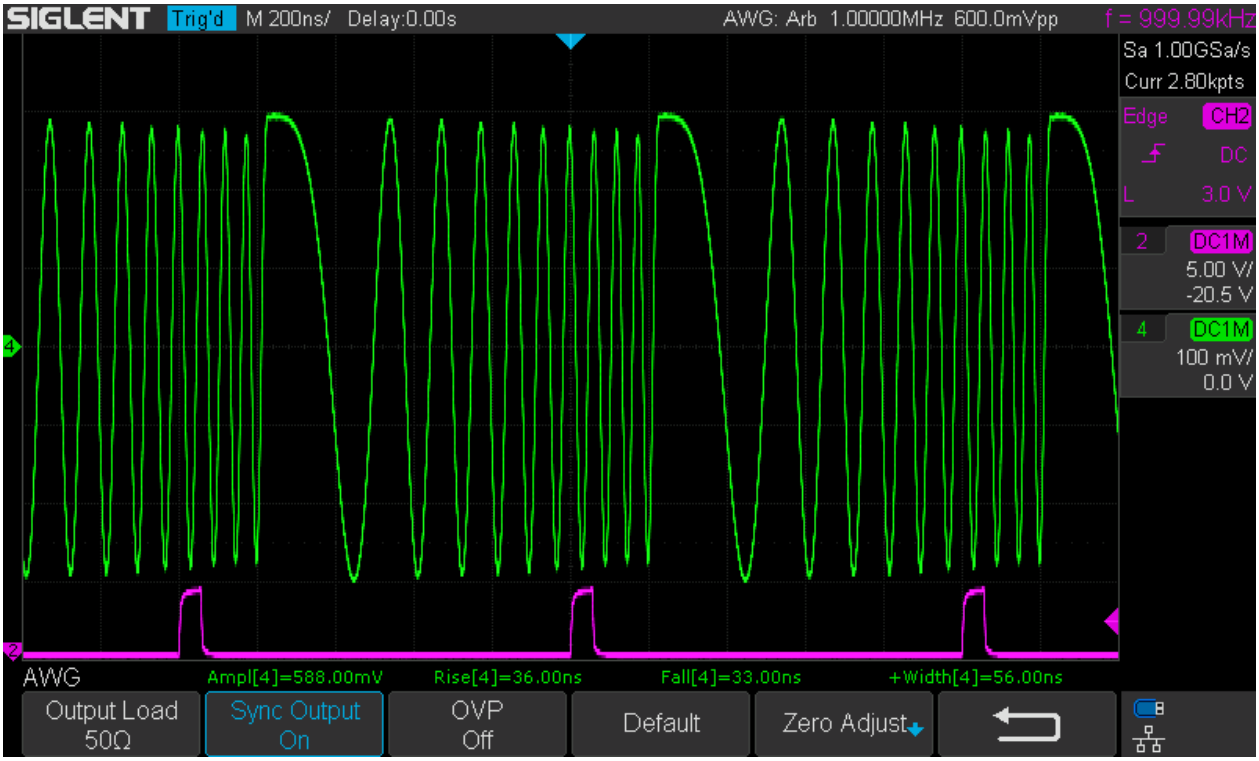
SAG1021_Arb_Int_Ppulse_5MHz

Other internal arbitrary waveforms are more complex and we'd need to set up some advanced trigger and/or use HF-reject or trigger hold-off in order to get a stable picture. A much easier way is to use just the trigger output of the SAG1021 as a trigger source (remember that this has to be enabled in the *Settings* menu first).

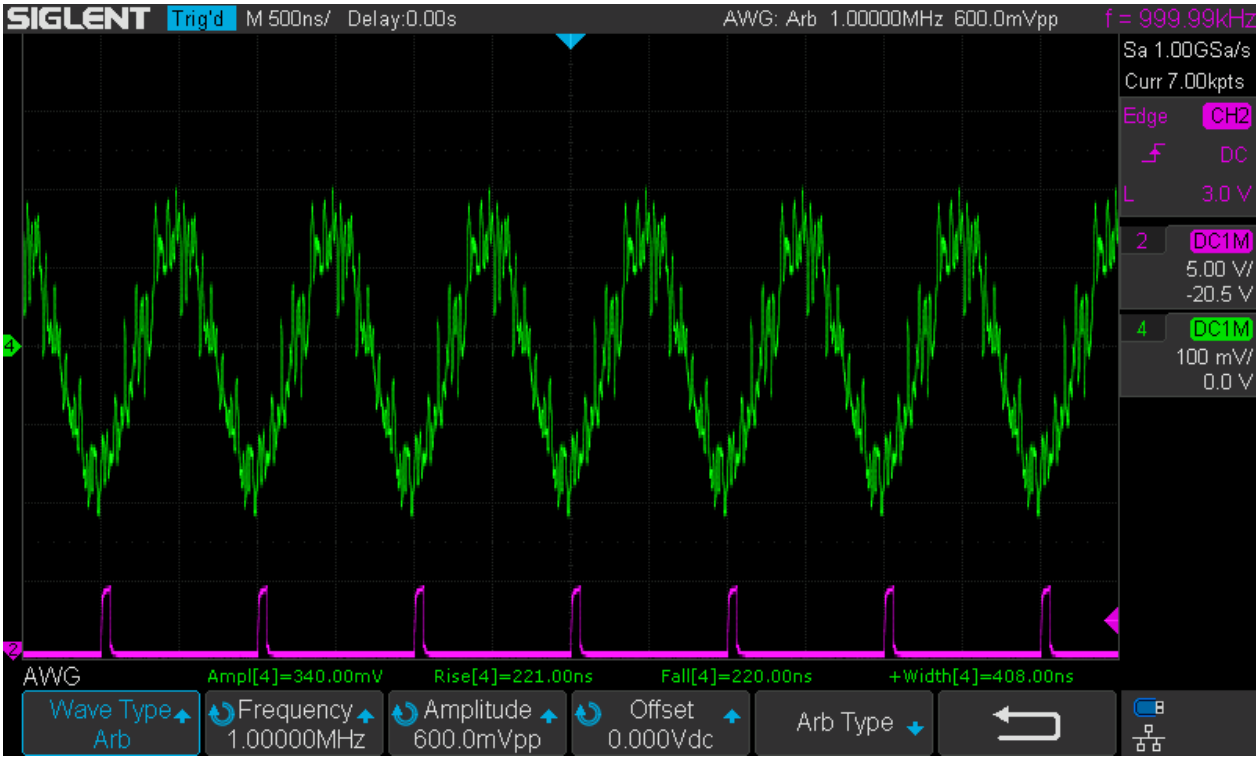
In the first following screenshot, the rigger output of the SAG1021 is fed into Ch.2 of the SDS1104X-E to get a stable trigger on the internal arbitrary Chirp waveform.

The same applies to the 2nd screenshot below, where the internal arbitrary SNR (Signal to Noise) waveform is shown. This simulates a noisy signal, but of course it would be much more versatile to do this with a dual channel function generator by just mixing and thus superimposing the noise signal from one channel on whatever signal the other channel puts out.

What we have learned by now is that the arbitrary function in the SAG1021 provides a rise time of <8ns, which is good news, particularly as this rather important information is missing in the data sheet.

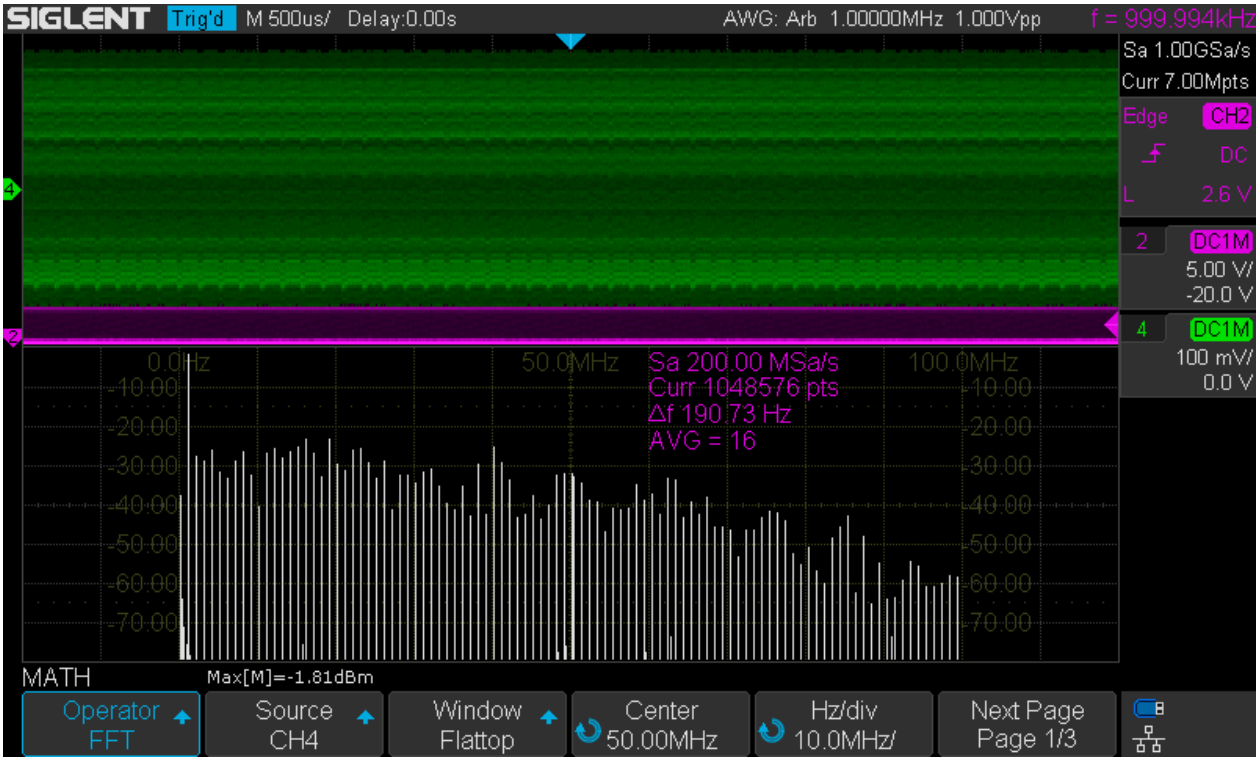


SAG1021_Arb_Int_Chirp_Sync



SAG1021_Arb_Int_SNR_1MHz

It would be interesting to see what the SNR wave looks like in the frequency domain, so let's just compute a FFT on it. The result can be seen below:



SAG1021_Arb_Int_SNR_FFT

The signal frequency of 1MHz sticks out at about -1.8dBm and the noise produces a number of discrete spectral lines with fairly constant amplitude up to some 25MHz with a very soft roll-off beyond.

Note that the power level of 1Vpp into 50 ohms would be +4dBm and if this were the amplitude of a clean signal with added noise the total level would rather be increased. But in this case, the noise is part of the waveform and the peak to peak amplitude already includes this artificial noise. As a consequence, the amplitude of the clean signal would actually be just 514mVpp.

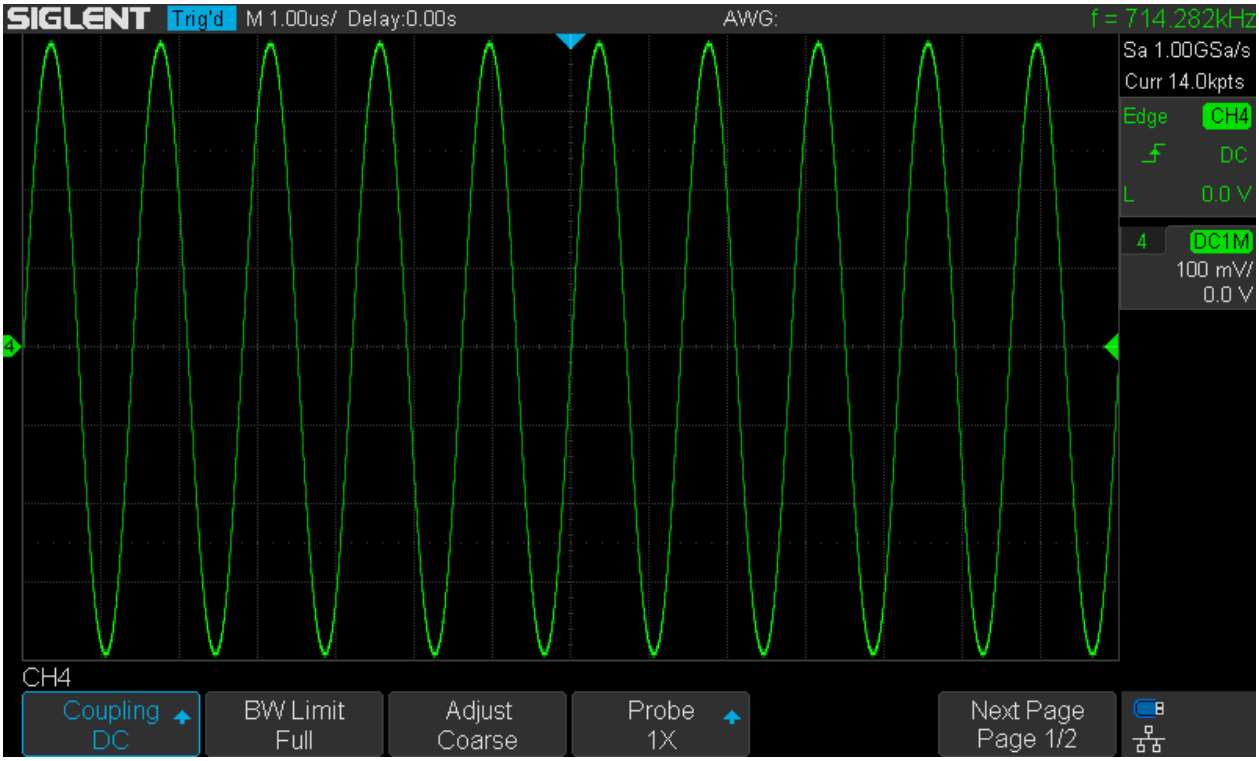
External

External arbitrary waveforms up to 16kpts can be temporarily loaded into the SAG1021. One could prepare an USB stick with a library of user specific arbitrary waveforms if needed.

There is Siglents free EasyWave application to create arbitrary waveforms and also read, write and copy them from and to external devices such as AWGs and DSOs. Of course we don't need that program as long we don't want to modify a waveform and then can just use an USB stick to transfer waveform data between devices.

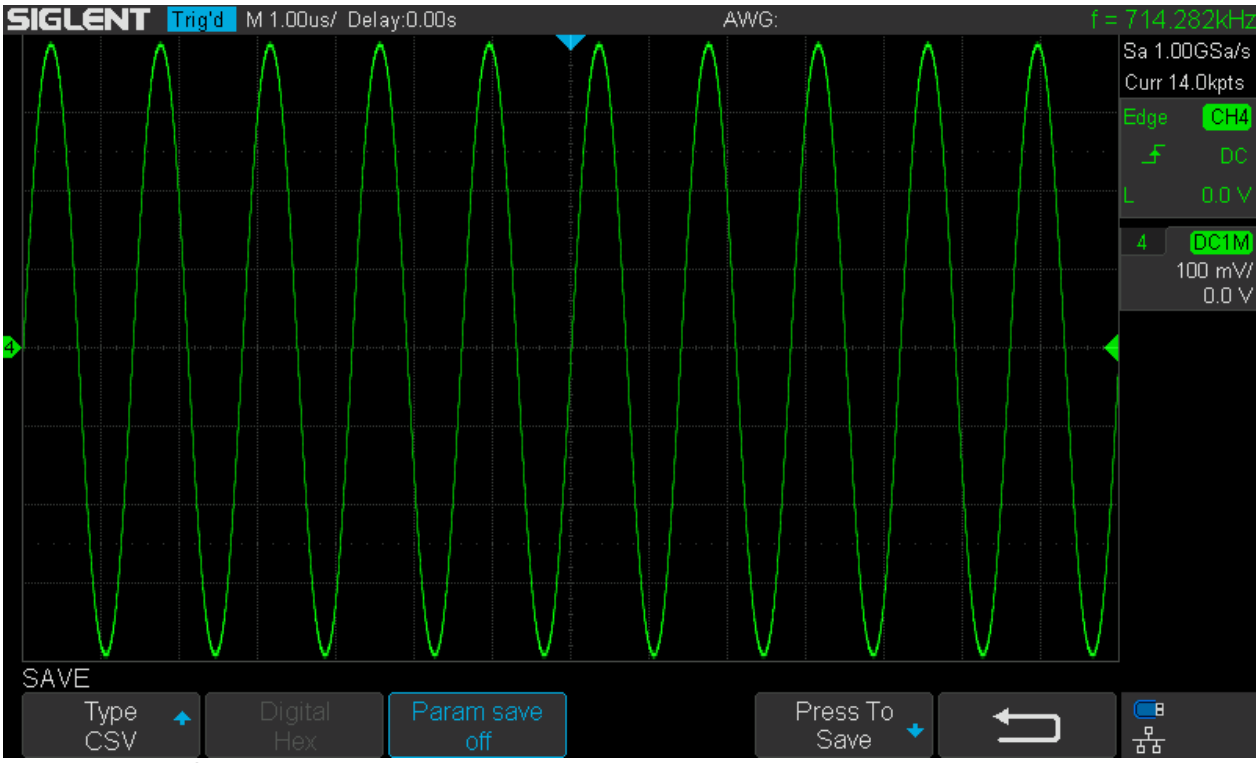
Let's look at the 2nd method first and load a waveform captured with the SDS1104X-E into the SAG1021. When doing this, we need to be careful to limit the record length to <16k, either by selecting 14kpts as the memory depth in the *Acquire* menu, or use a timebase fast enough so that the recordlength does not exceed 16kpts.

For this demonstration we take a sine wave with a period of precisely 1.4µs so that we get 10 full periods on the screen of the SDS1104X-E when it is set to a timebase of 1µs/div.



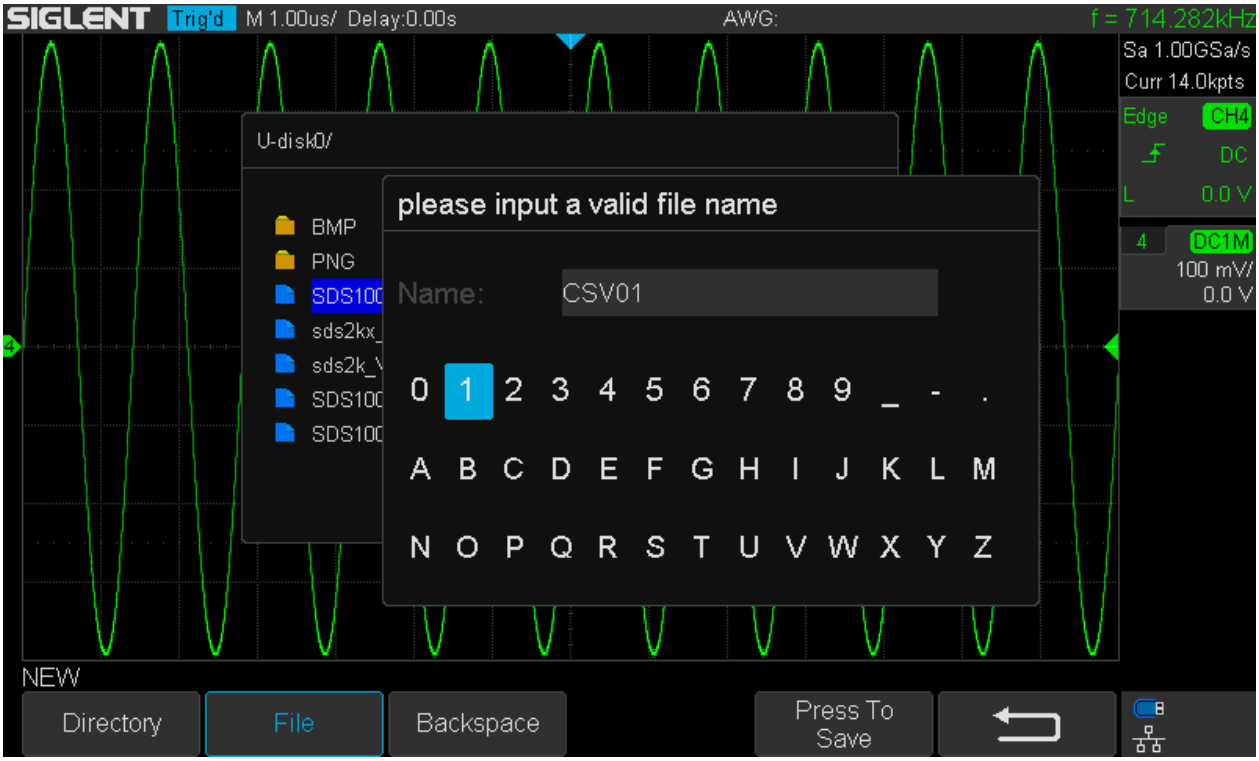
SDS1104X-E_Sine_P10_Capture

Now we want to store that waveform on an USB stick. In the SAVE menu, we select *Type CSV* and make sure that *Param save* is off.



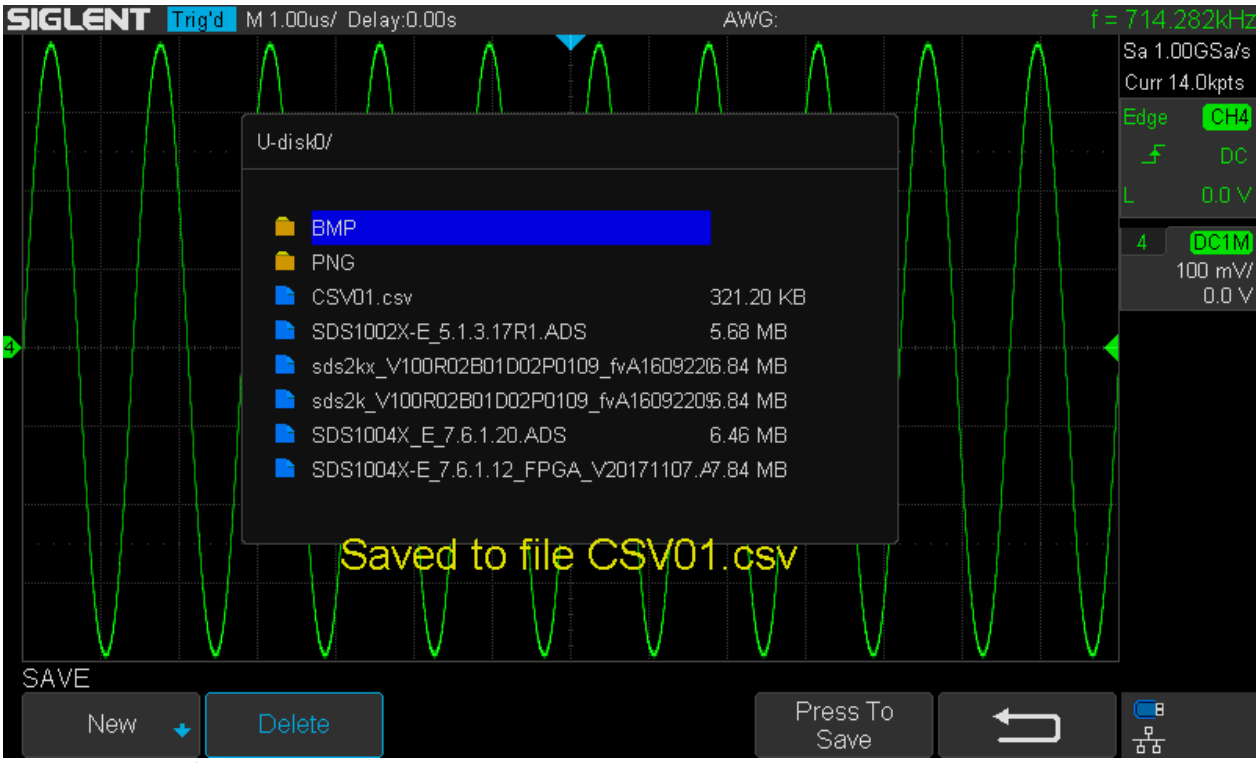
SDS1104X-E_Sine_P10_Save_1

Next we push the *Press To Save* button and enter a filename. Since this is not easy without keyboard, just a short number will do. We can always rename the files later on a PC.



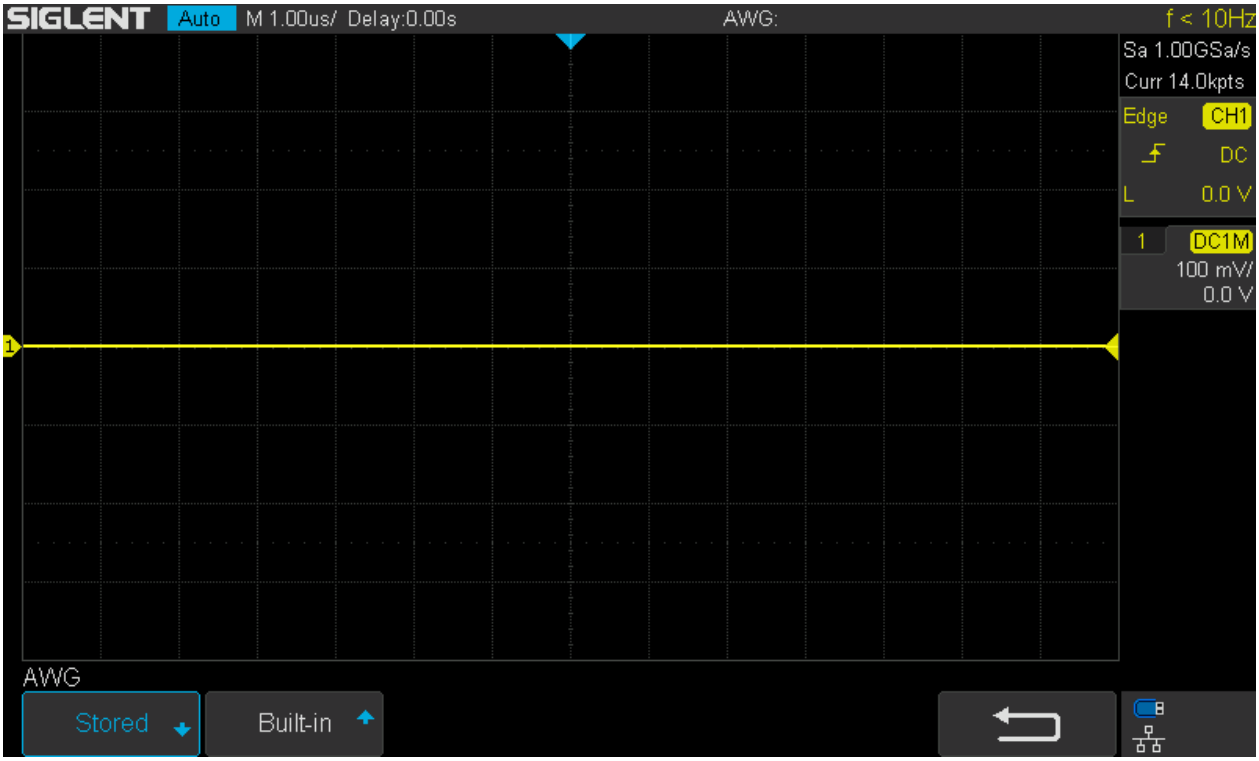
SDS1104X-E_Sine_P10_Save_2

Finally we push *Press To Save* in order to get the CSV data on the stick.



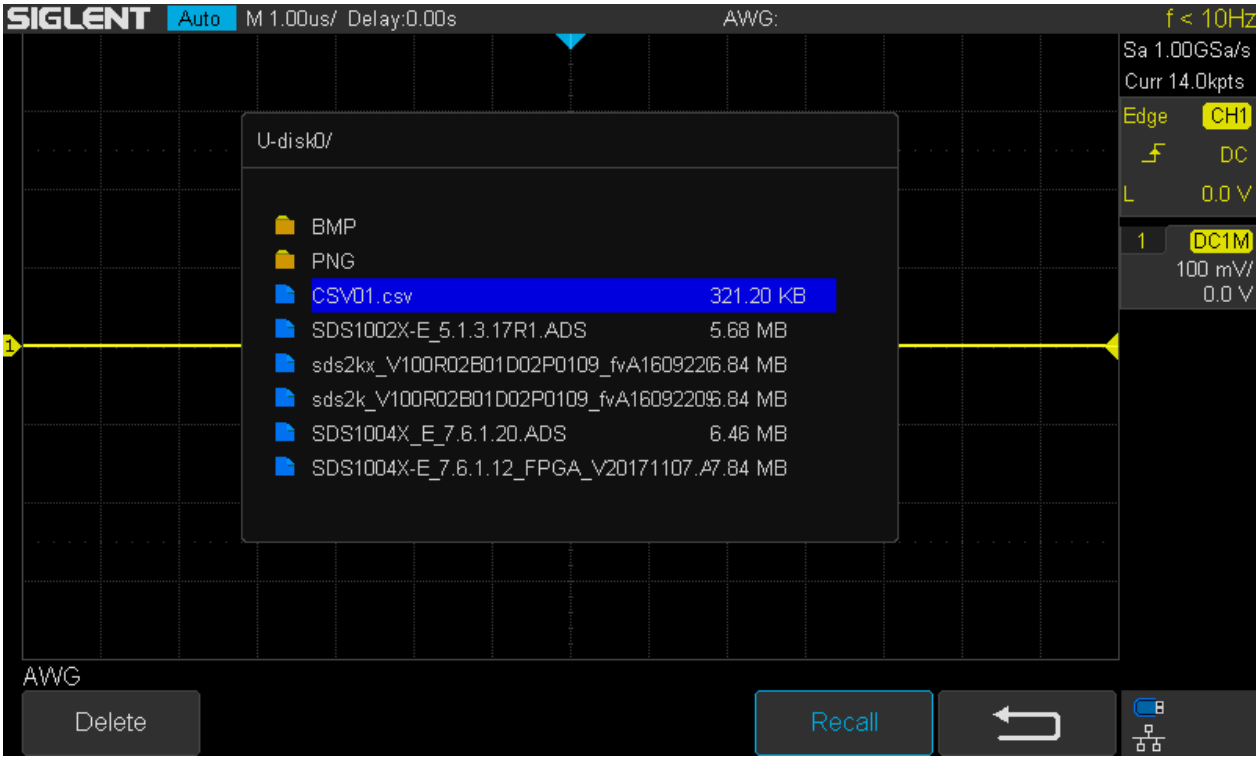
SDS1104X-E_Sine_P10_Save_3

Now we just need to load the data into the SAG1021 AWG. To do this, We select *Arb* as the wave type and then enter the *Arb Type* menu. There we need to access the *Stored* menu.



SDS1104X-E_Sine_P10_Load_1

Now we can see the contents of the USB stick and highlight the file we want to load by means of the universal control, then just press *Recall*.

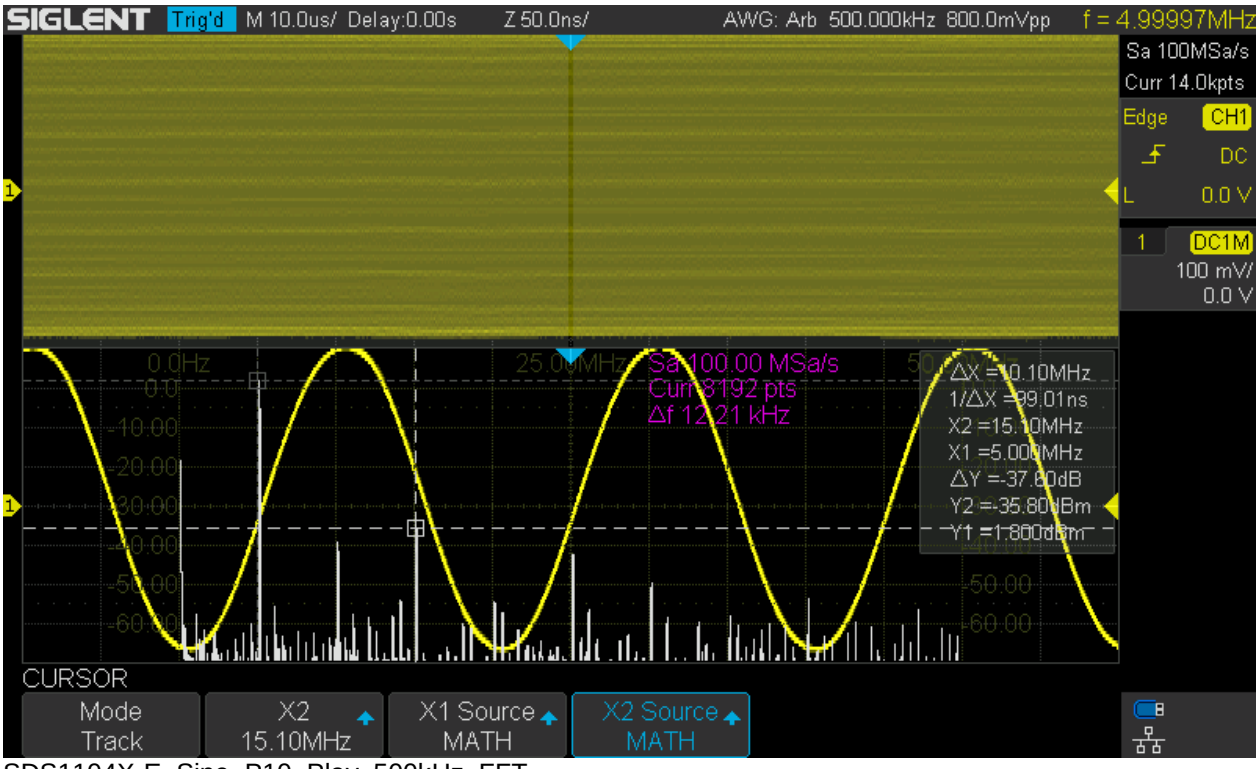


SDS1104X-E_Sine_P10_Load_2

Now we can play the transferred waveform at e.g. 500kHz and get a 5MHz sine output as expected.

SDS1104X-E_Sine_P10_Play_500kHz

How's the quality of that sine wave copy? Let's use the FFT to find out:

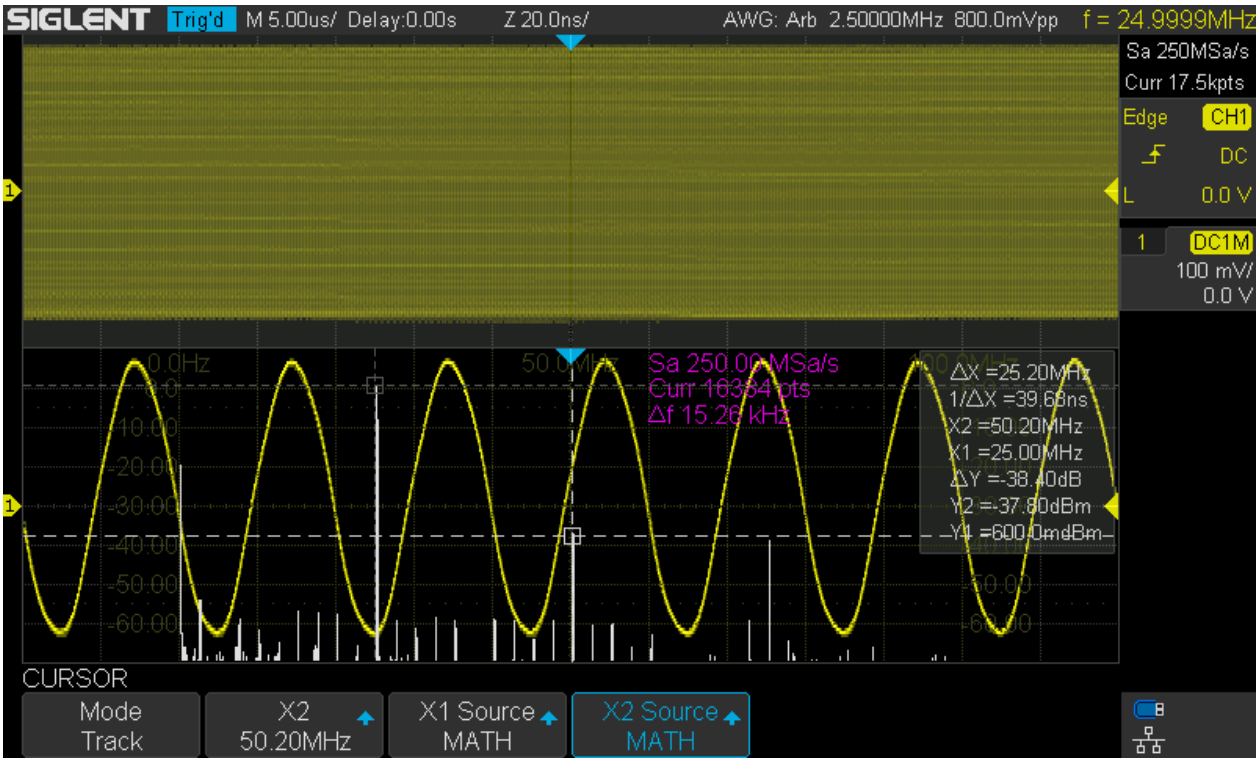


SDS1104X-E_Sine_P10_Play_500kHz_FFT

With only -37dB$_C$ for the 3rd harmonic the quality of this sine is a far cry from the original. This shouldn't come as a surprise though, as we now have less than 8 bits resolution, whereas the internal waveforms

use 14 bits. In general, this method of signal transfer is far from ideal; any decent function generator has at least 12 bits resolution whereas the DSO has just 8 (where only 200 codes are actually used for the full screen height and the rest is headroom), so a major loss of fidelity is inevitable.

Now we can try higher frequencies, like 2.5MHz in order to get a 25MHz output, just like the upper limit for the internal sine.



SDS1104X-E_Sine_P10_Play_2.5MHz_FFT

We still get -38 dB_C for the 3rd harmonic, which is even a tad better than for 500kHz. So we can actually use this technique to increase the max. output frequency of arbitrary mode. But that's to be expected, as many internal arbitrary waveforms contain higher frequency content as well.

We should avoid higher frequency components above 25MHz though, as the signal quality degrades quickly and spurious signals will emerge.

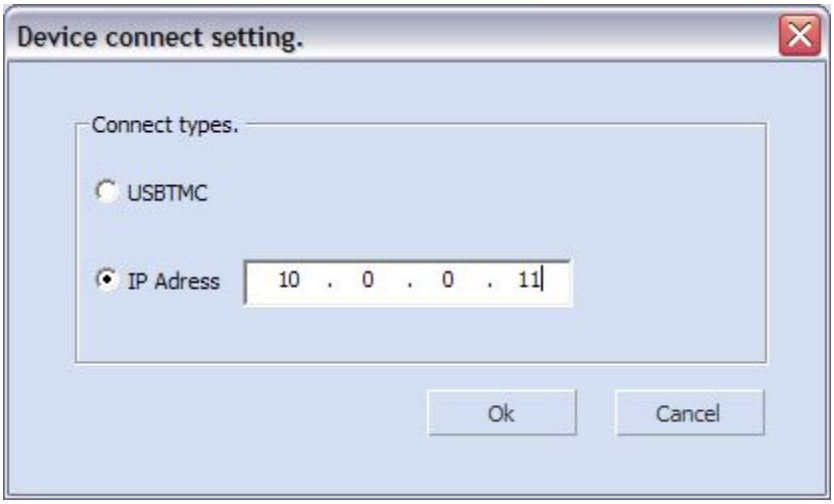
Now let's have a look how to do it using the EasyWave program.

First I'd like to say that I am pleasantly surprised about what we get here for free. Of course it's not a high-end program, the English translation is a bit clumsy at times and it does not even have a title, but it appeared stable and everything I've tried worked fine first try. I have not tried to edit waveforms, but it seems to have all the necessary tools, including predefined waveforms, drawing tools, equation draw, coordinate draw, grid, tracking cursors and zoom.

Even more impressive, importing a waveform from the SDS1104X-E worked first try and was absolutely easy. I needed not even worry about the record length, as the required downsampling to 16kpts was quite obviously done automatically by the import tool.

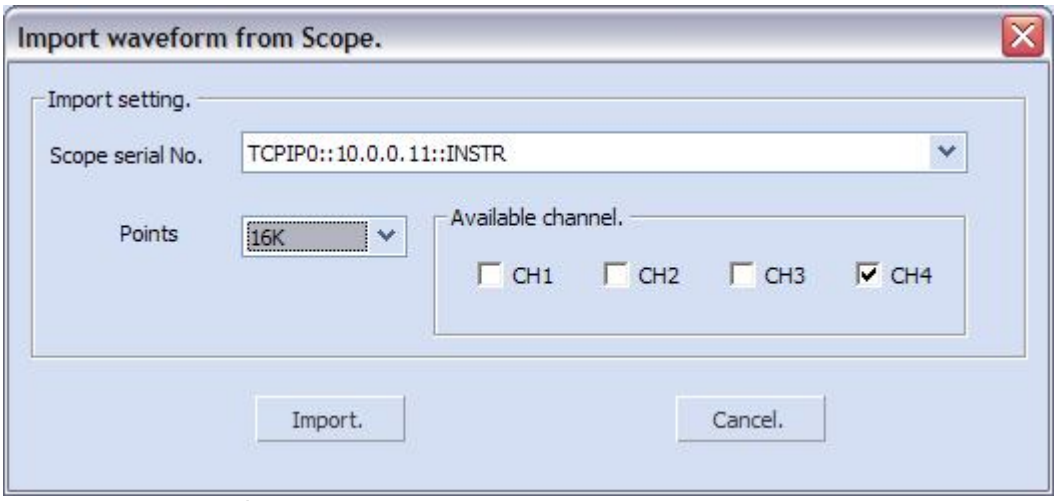
For this test I've fed a PRBS signal from some other generator into channel 4 of the SDS1104X-E and attempted to capture this waveform from remote with EasyWave.

We first need to establish a connection to the device, the SDS1104X-E in this case, but it could be a signal generator as well. For doing this, we have the choice between USBTMC or LAN, which I have used. First we need to enter the IP address:



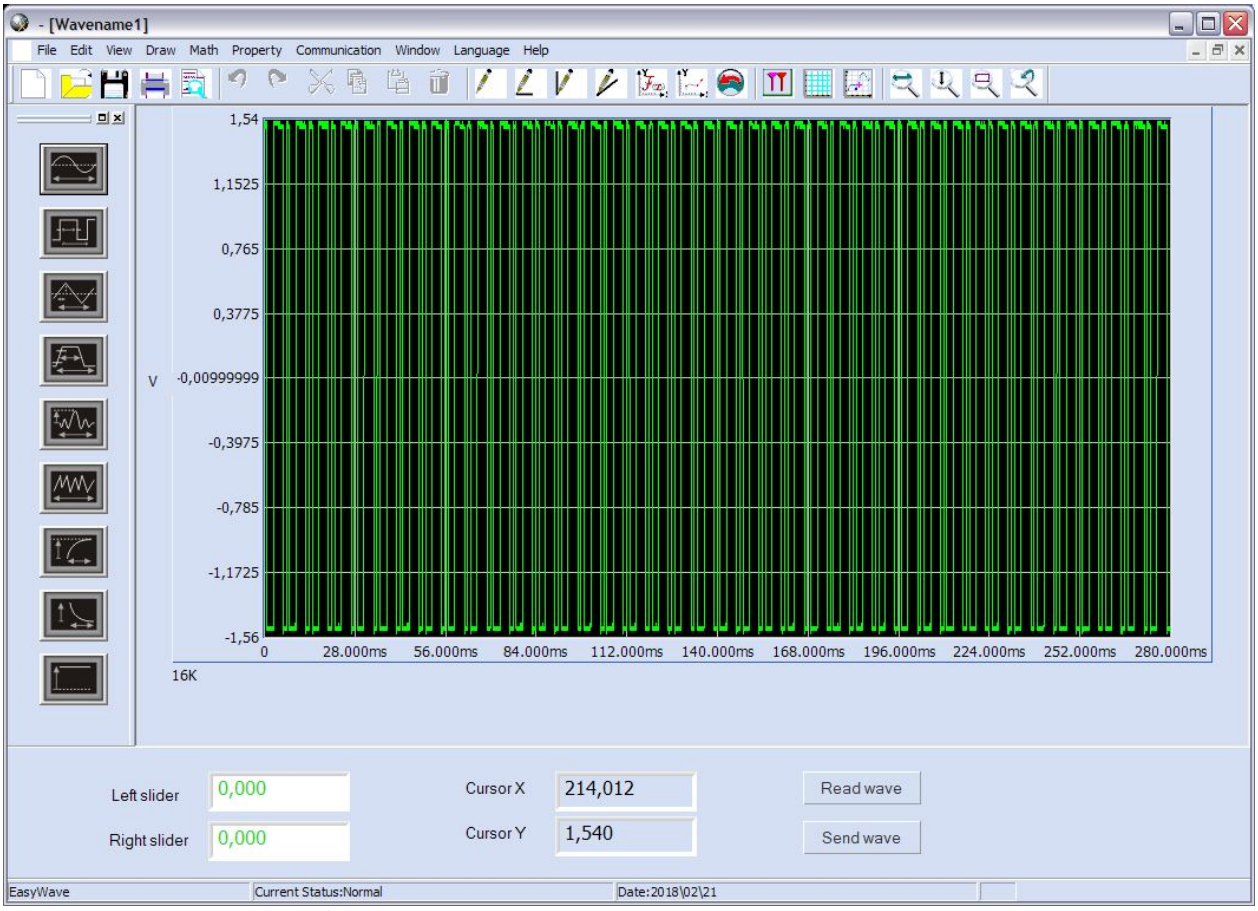
EasyWave_Connection_1

Next we select the record length which is set to 16k by default already and the scope channel we want to import from.



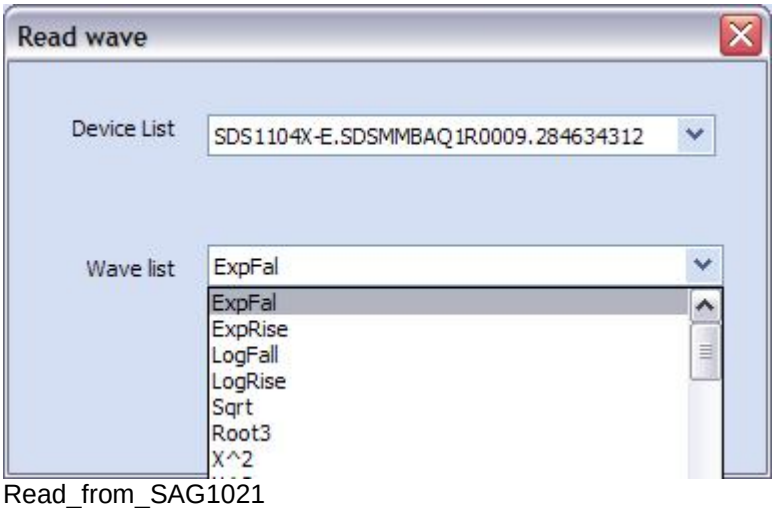
EasyWave_Connection_2

Lo and behold, the waveform appears with the correct amplitude and time scale. Victory!



EasyWave_PRBS_Import_from_SDS1104X-E

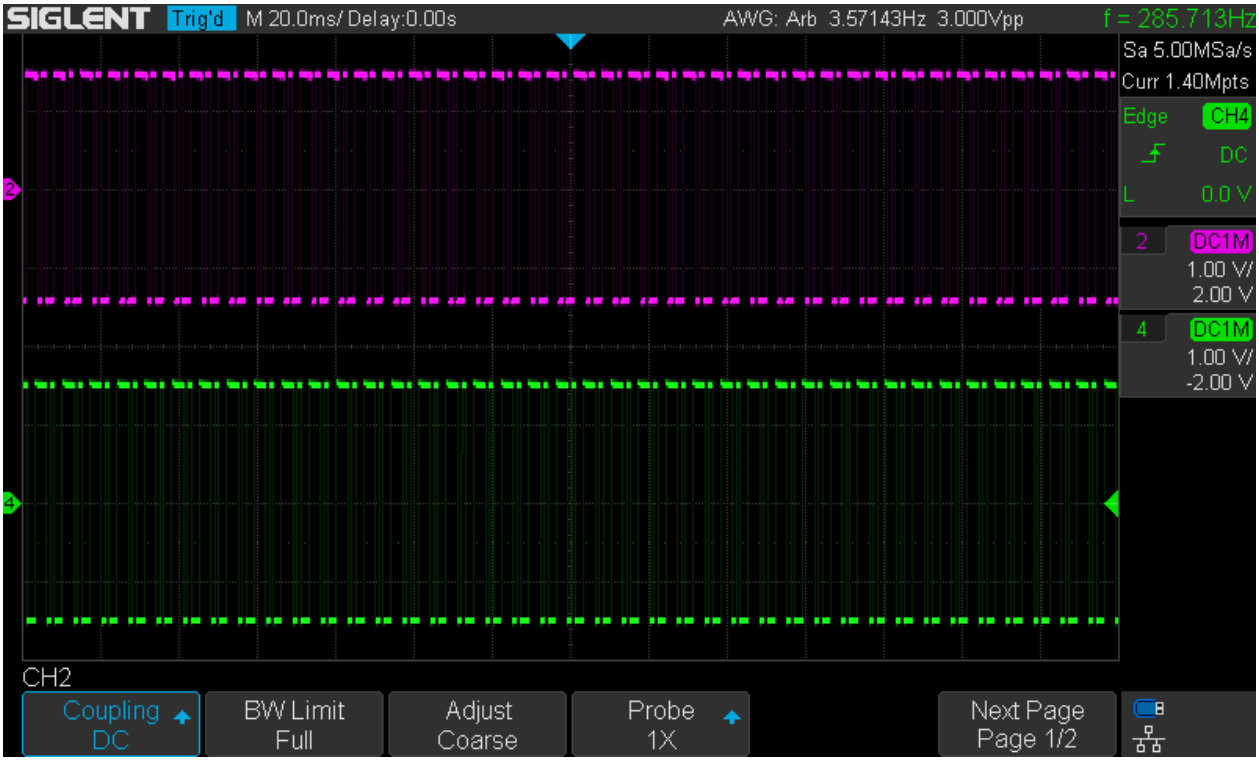
We cannot store the waveform into the SAG1021 directly, even though we can actually access it through the scope and could even read its internally stored arbitrary waveforms.



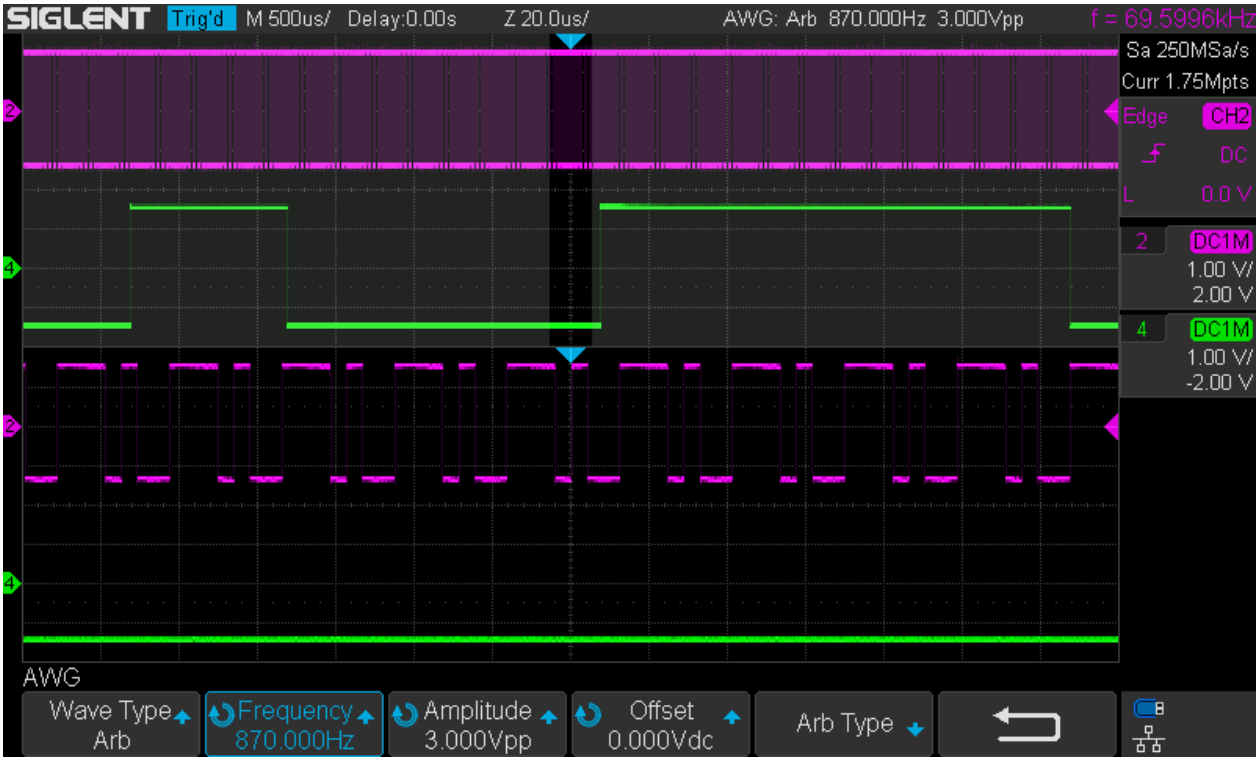
Read_from_SAG1021

There is just no writeable permanent memory in this device, so we need to use an USB stick to store the waveform on the PC.

I've used the default name and just clicked on "Save". Then plugged the USB stick into the scope and loaded it into the SAG1021 just like in the first example. As a result, I could now see both signals, the original one on Ch.4 and the copy from the SAG1021 on Ch.2 and of course it can be replayed at an arbitrary speed…



SAG1021_Imported_Arb



SAG1021_PRBS