Using the AARONIA RTSA Spectrum-Condition- and Trigger-Blocks

General Concept

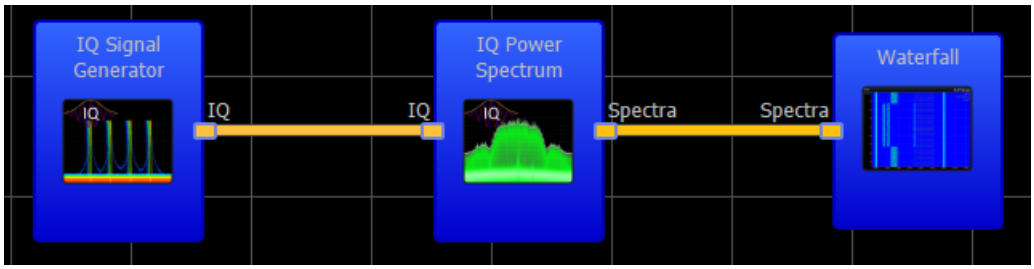
Each sample (may it be an IQ sample, audio sample or spectrum) may carry additional meta data. This meta data includes up to four Boolean flags (C0 to C3), that can be set using a condition block, visualized in the waterfall view or used as a trigger source.

The condition flags are also stored in RTSA files and can thus be used for a later analysis.

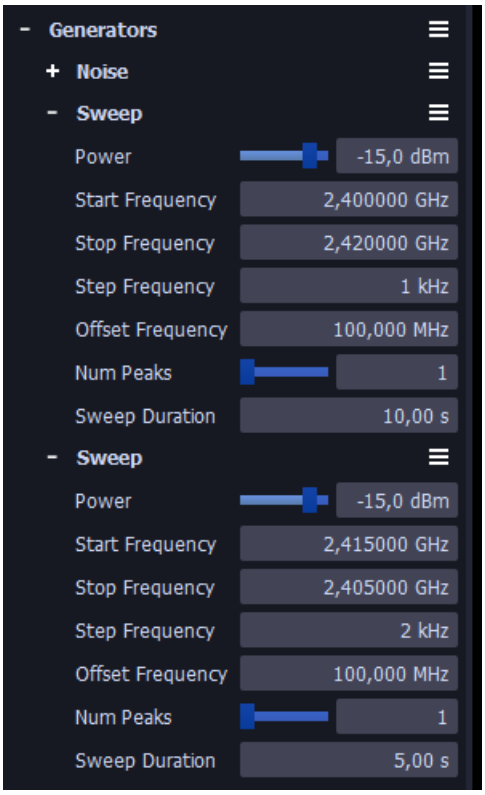
Simple Event Sample

Sample Setup

We use the “IQ Signal Generator” block to create two frequency sweeps.

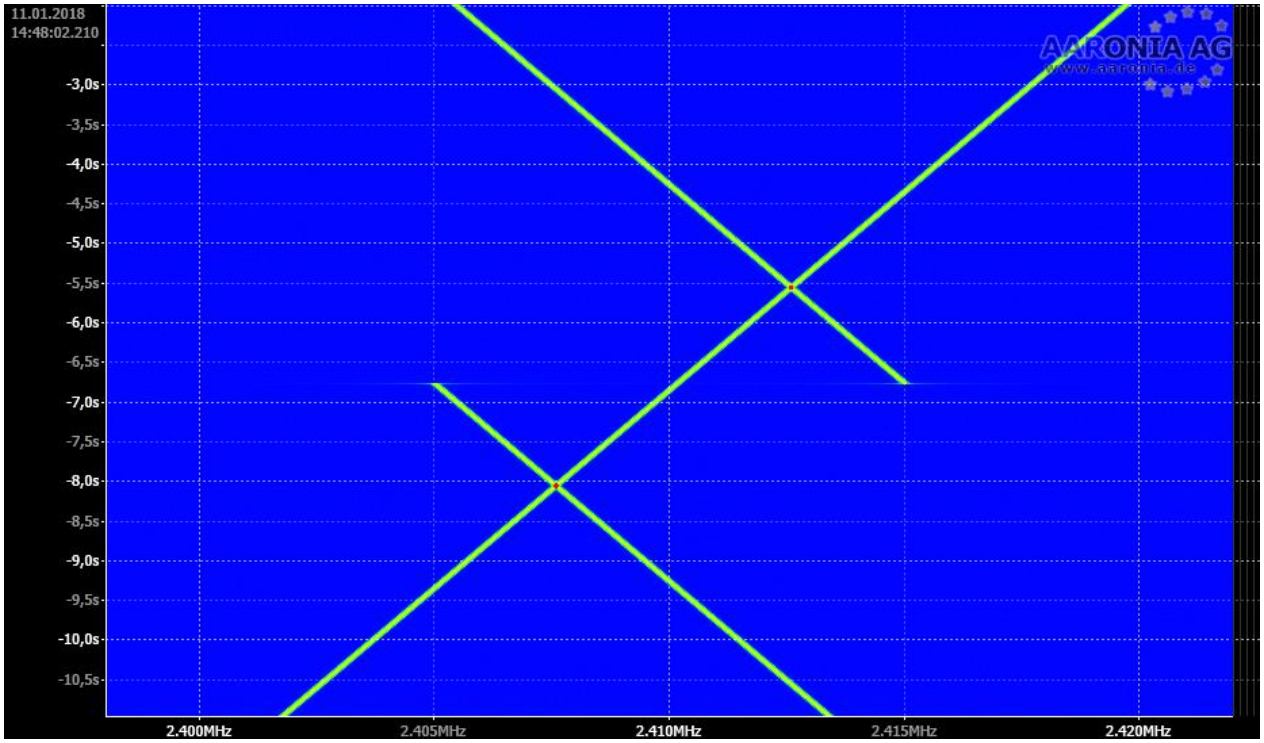


The first sweep will move from 2.4GHz to 2.42GHz in a 10second interval, the second sweep in reverse direction from 2.415Ghz to 2.405GHz in a five second interval. We also add some noise to keep the setting interesting.



We then feed the IQ stream into a IQ Spectrum Block to generate series of power/frequency spectra for further processing.

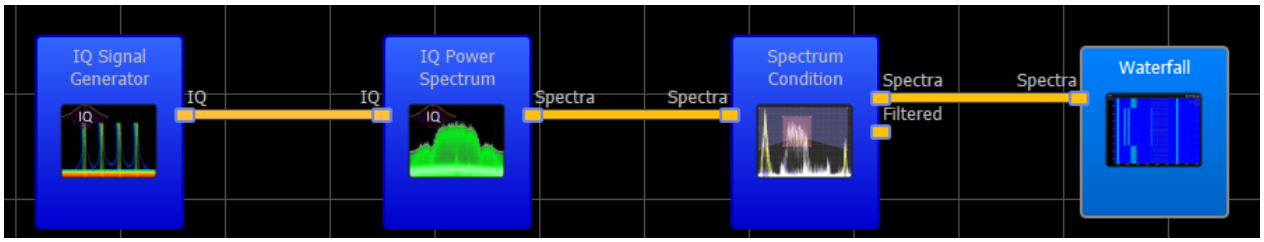
The sweeps will meet repeatedly at two different frequencies and generate a little spike there.



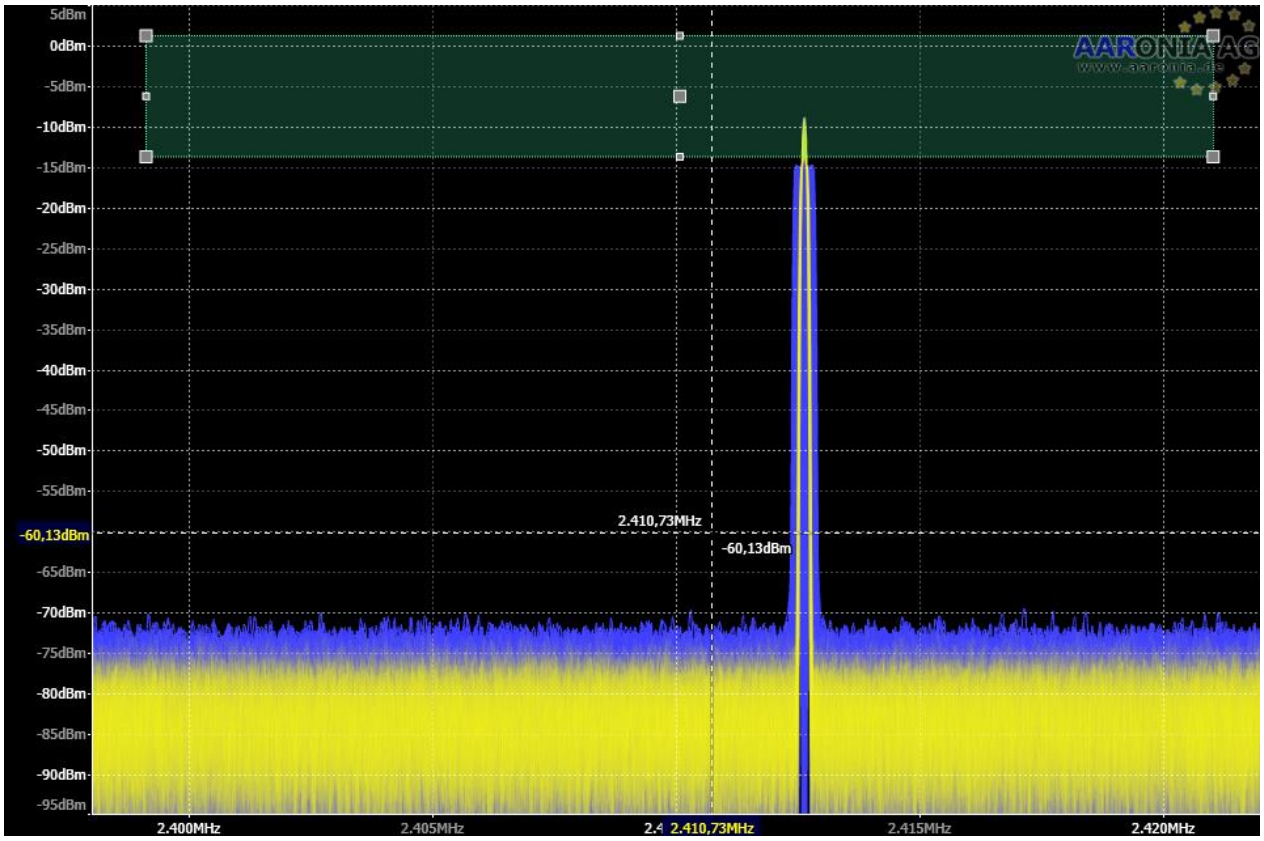
This example will try to capture the rendezvous of the two sweeps.

Spectrum Condition

We insert a spectrum condition block to set the condition flags based on the power level of the signal.



We mark the area of interest with the cursor and add a trigger area using the menu in the condition block settings.



The preview will highlight all spectra that meet the condition with a yellow color (you can change this in the view config area).

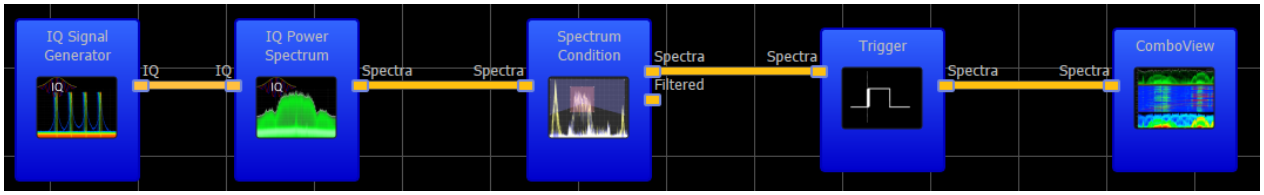
We use the C0 flag as target channel and set it to true, whenever at least 0.1% of the spectrum hits the condition area. We can limit this by percentage of spectrum and minimum number of consecutive samples. The operation may use the incoming flags using a Boolean operation thus allowing the cascade of several conditions.

We can now observe the flags in the waterfall view:

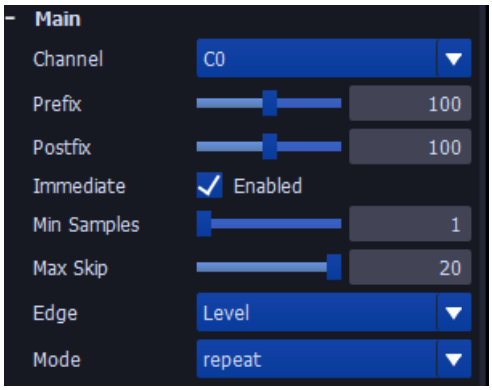
A time compressed view will always show the logical or of the conditions. The event itself is too short to easily capture without compression, therefore we will use a trigger block.

Extracting Events using the Trigger Block

We add a trigger block and switch from the simple waterfall to a combo view.



The purpose of the trigger block is to extract temporal ranges from a sequence of samples using one of the condition flags previously set by a condition block.

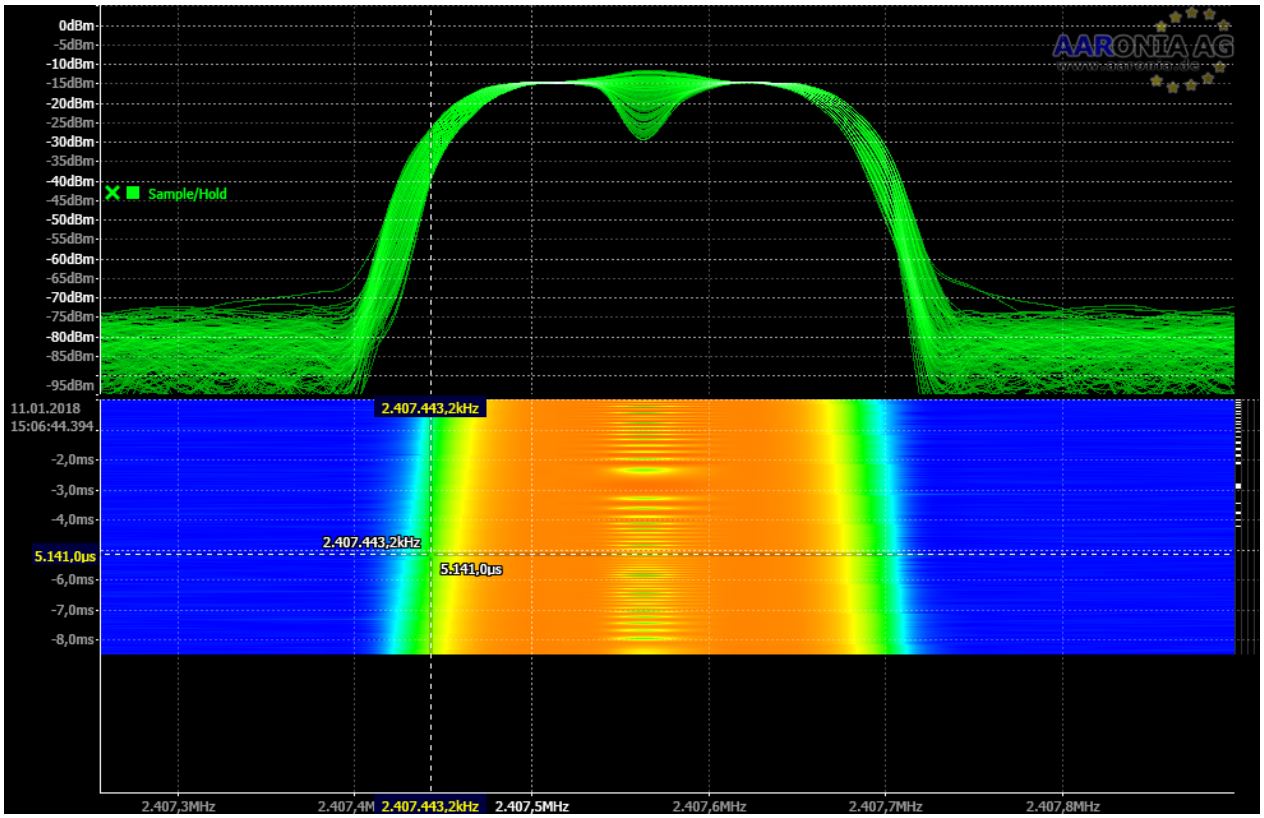


The trigger event is controlled by:

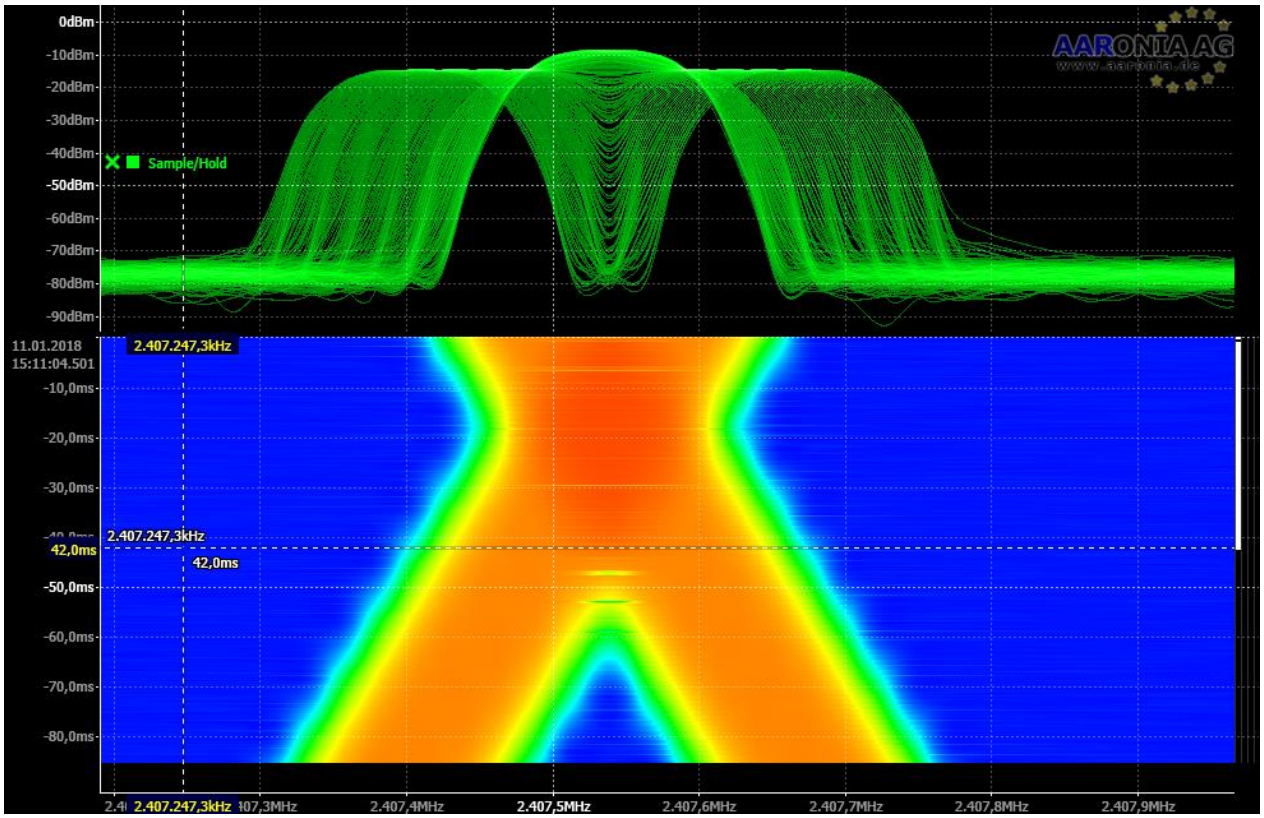
- Channel: The flag to query
- Min Samples: Number of consecutive samples that need to have the flags set
- Max Skip: Maximum size of a range of samples in a sequence that have the flag not set
- Edge: Trigger edge, either rising, falling or level
- Mode:
    - Repeat: Trigger again and again, sending each trigger sequence as an independent stream
    - Continuous: Concatenate all trigger sequences into one stream
    - Once: Trigger only once – changes to Off after the event
    - Off: Don't trigger anymore

Capturing a single event

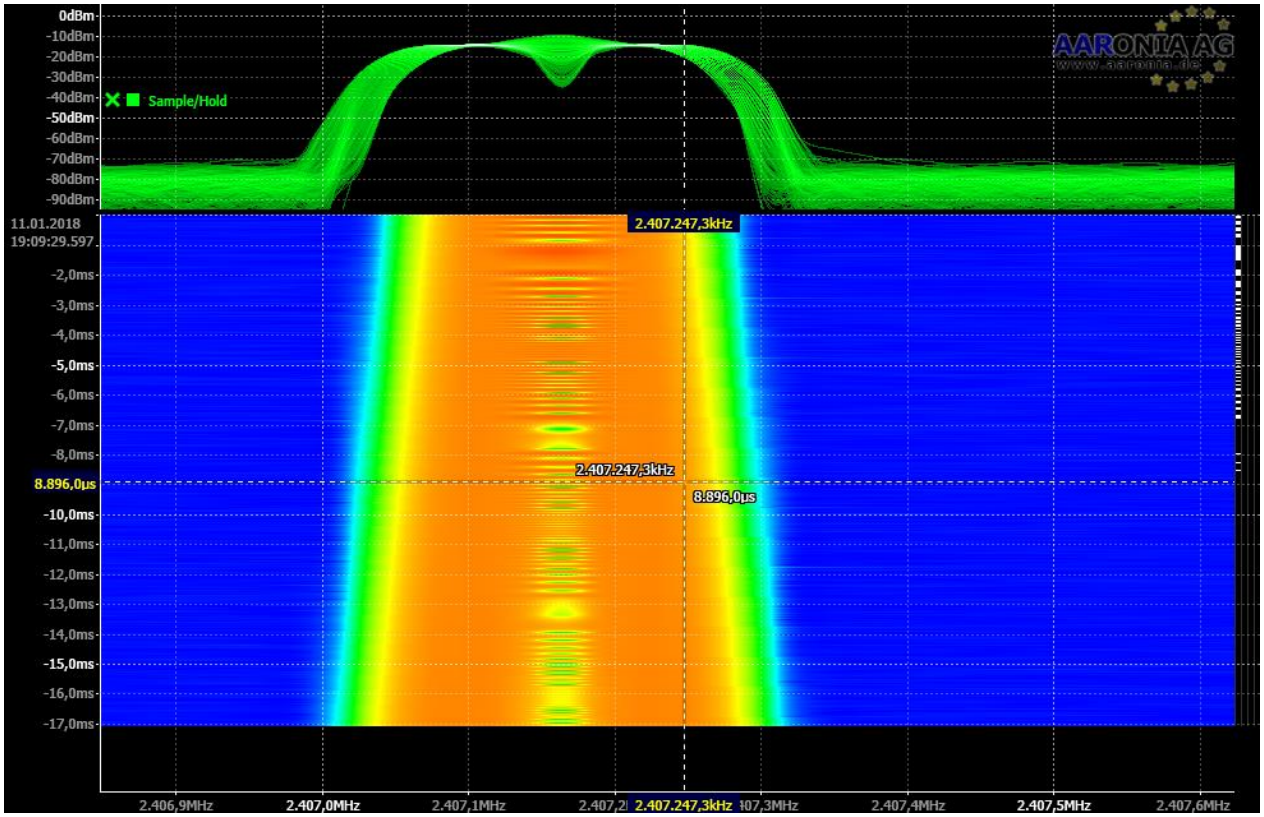
The first experiment will use "Riding Edge" and mode "Once":



We can observe the moment, when the two sweeps meet. The trigger event is in the center of the history range because we have requested the same amount of prefix and postfix samples. If we increase the number of samples requested, we get a longer snapshot:

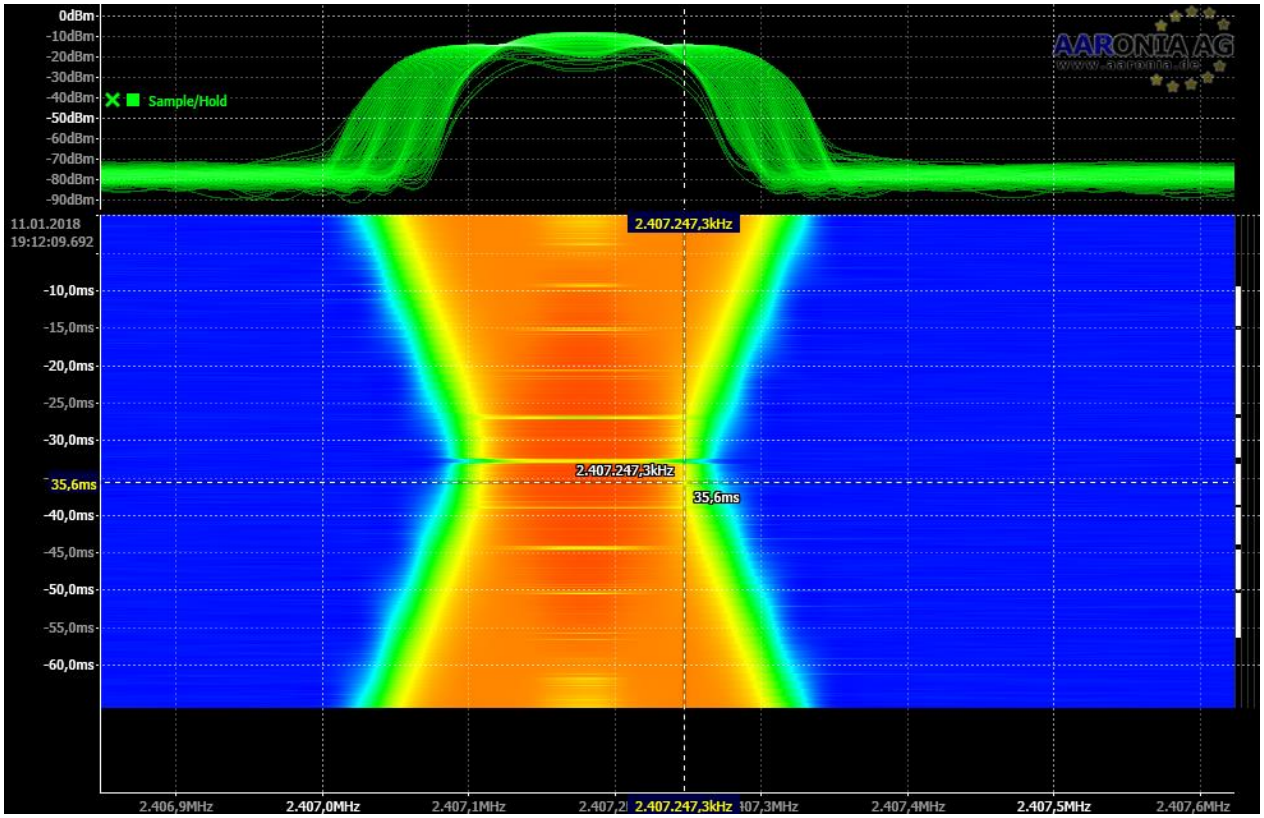


If we want to capture the full sequence, we need to switch from “Edge” to “Level”.



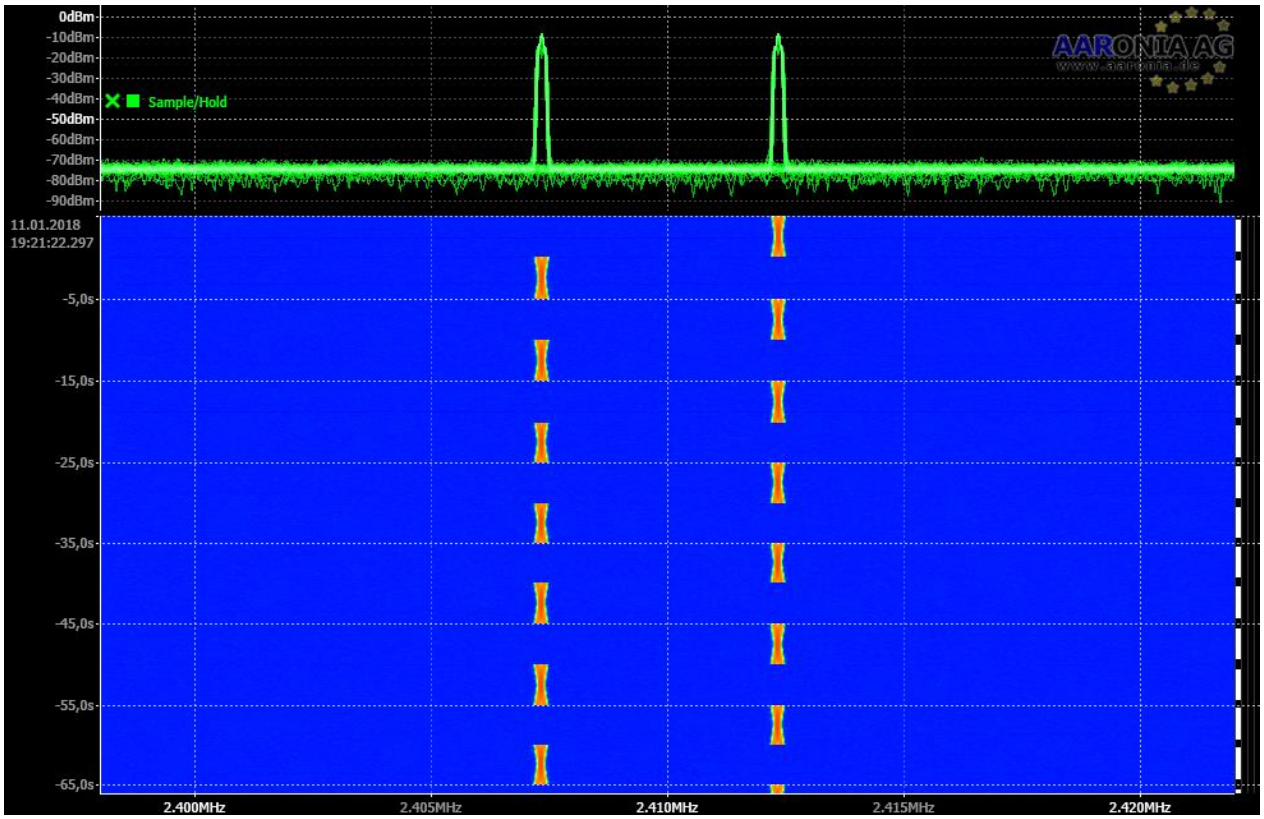
The result is not as expected, and it can be clearly seen why. The combined signal strength fluctuates due to the phase offset between the two signals. The trigger thus resets immediately after the first peak and only captures the suffix.

The “Max Skip” setting allows us to specify the maximum length of a period without the condition set, that is ignored.



Extracting Multiple Events

The trigger mode “continuous” allows us to capture all trigger events. This way we can reduce the stream to the areas where a condition happens, as well as some prefix and postfix samples.

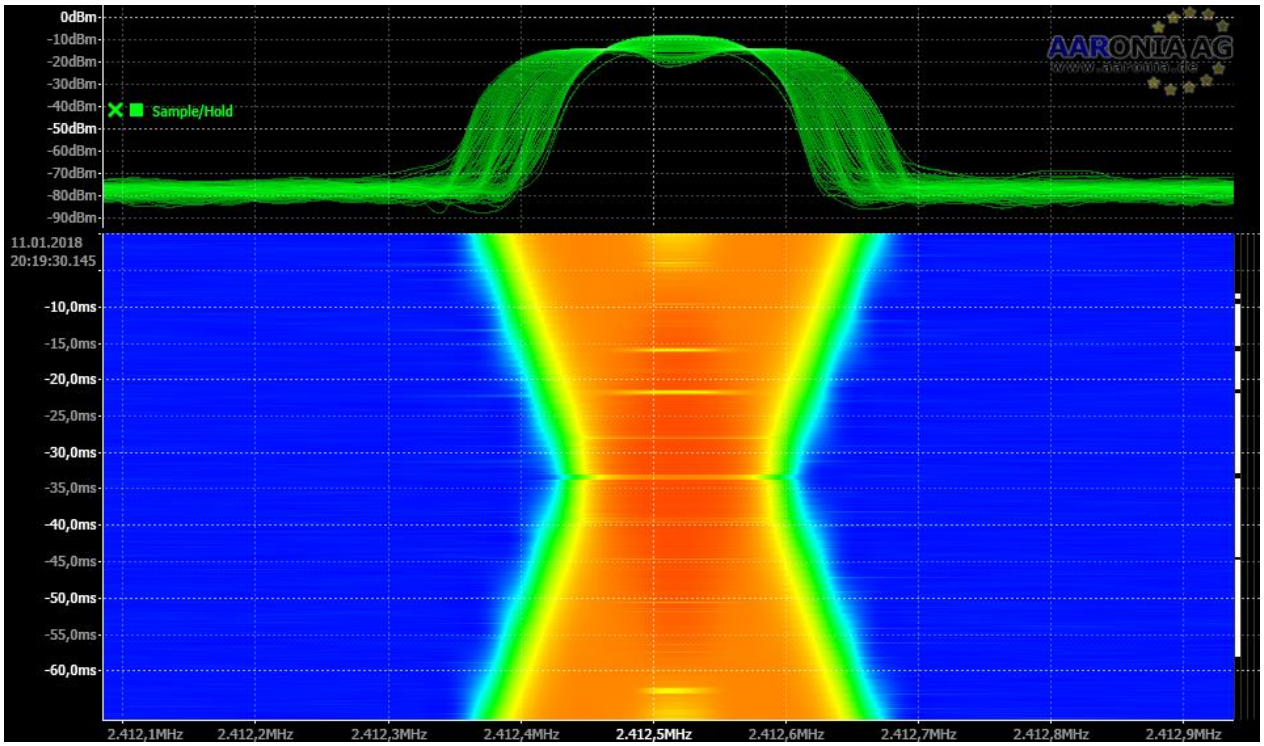


This mode is most suitable for extracting a series of events for e.g. storing in a file, or accumulating signal power of a sparse and noisy signal.

Continuous Monitoring of Events

The “repeat” trigger mode captures single events, but does not stop after capturing, so you will get each individual event as a stream.

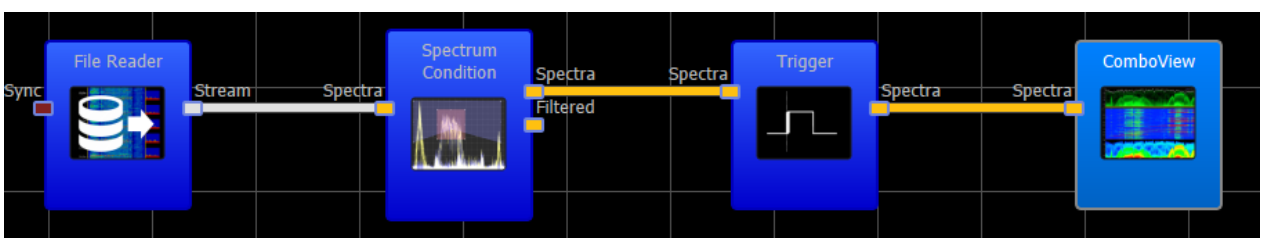
The output of the waterfall can be scaled to a single event using the “dynamic full range” checkbox in the waterfall.



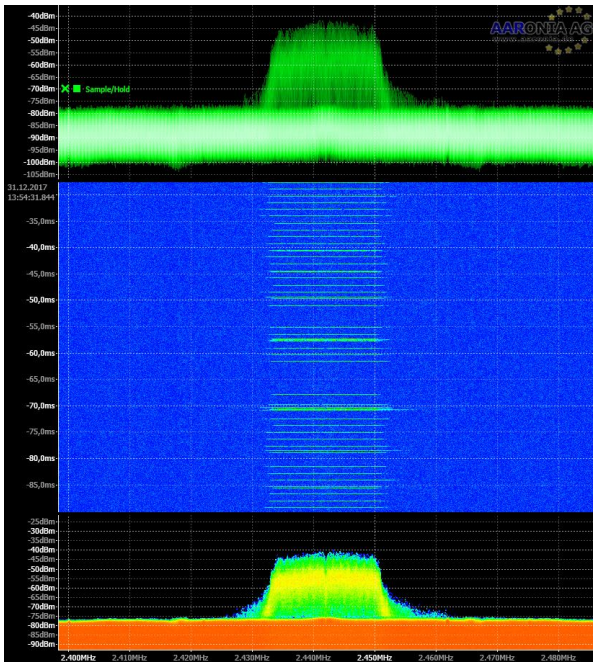
This way we can monitor a live signal and just view the samples that satisfy the event condition.

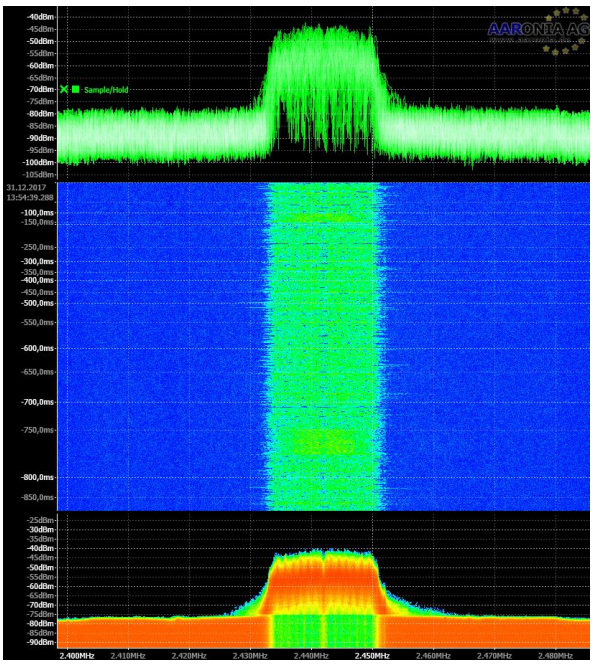
Real World Example

We will now look at a real-world example. We have a capture of a WiFi channel, and we want to reduce it to a series of longer bursts. The data is captured into a file, so we use a file reader block, a spectrum condition block, a trigger block and finally a combo view block.

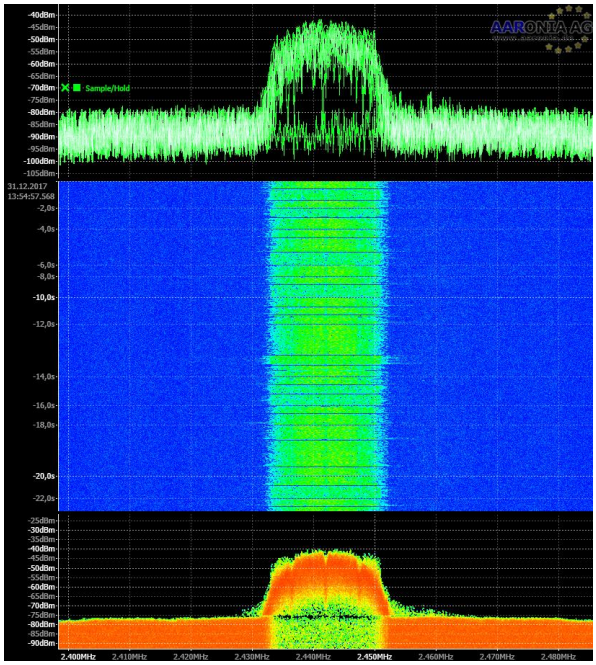


Even though the WiFi channel is quite busy, there is a lot of unused time see image on the left. We can filter out all the idle time using edge filter and no prefix, postfix.



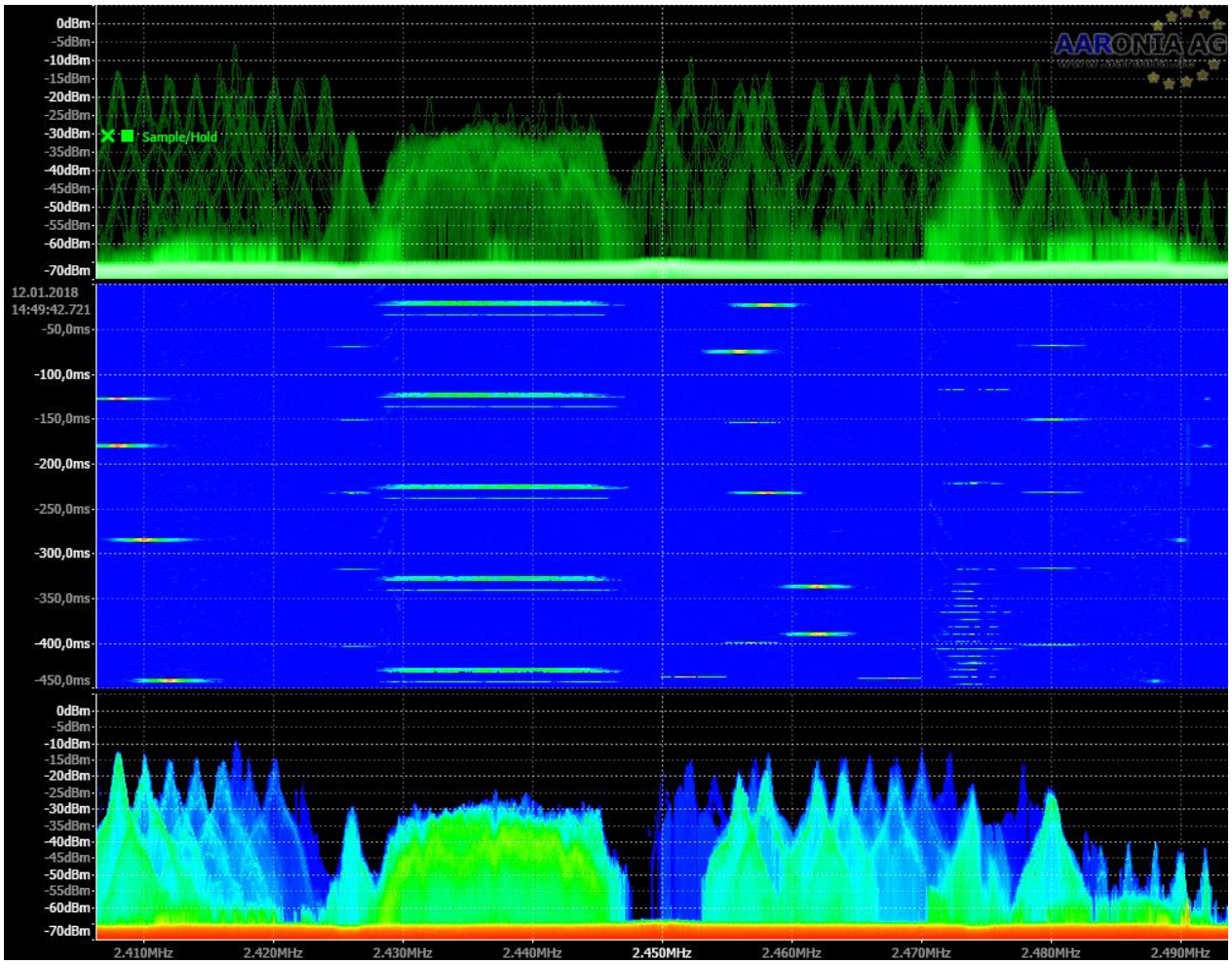


We can also try to capture only longer sequences by setting the Min Samples. Using a prefix of one helps us to see the boundaries.

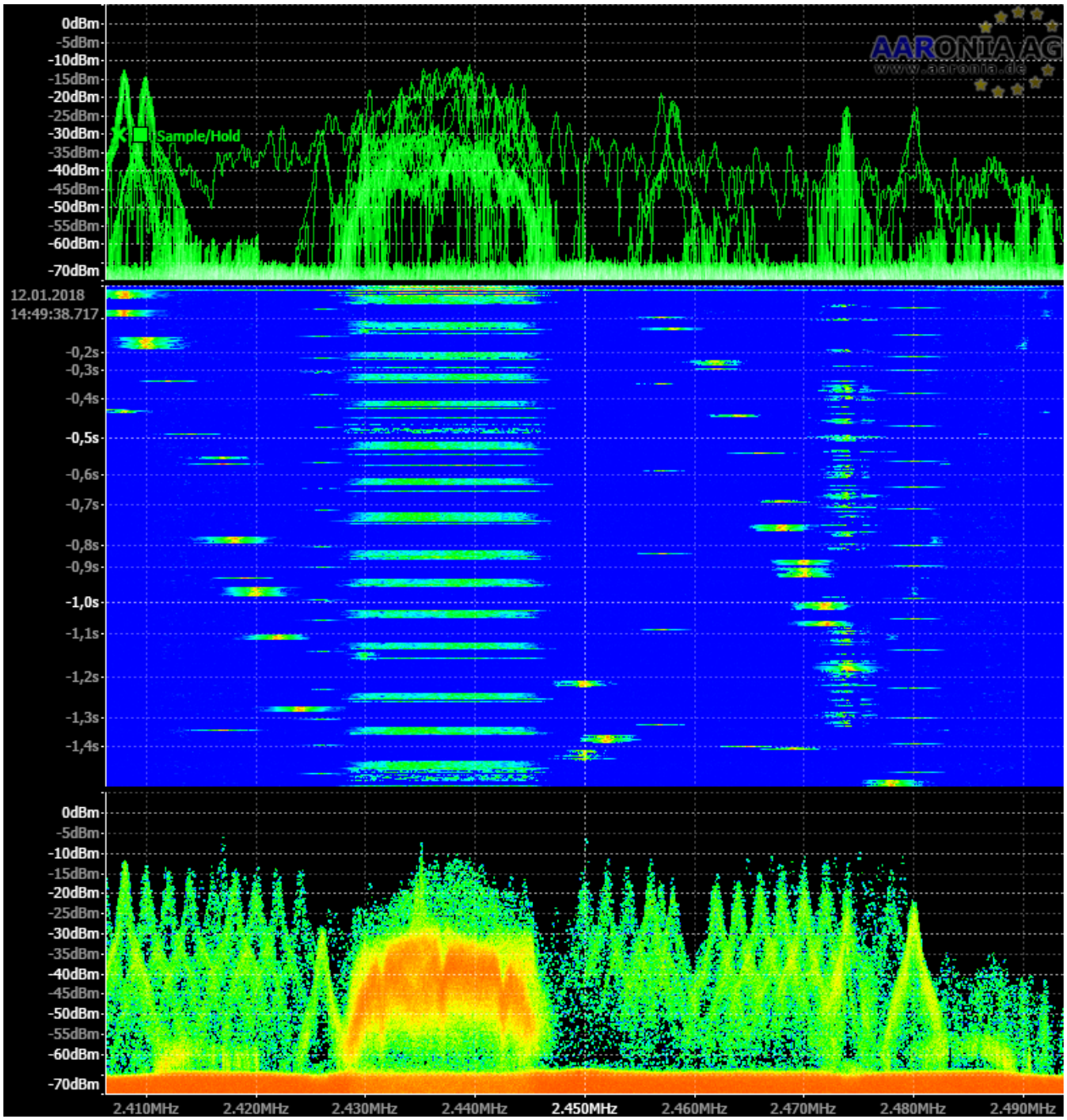


Extract Bluetooth Signals

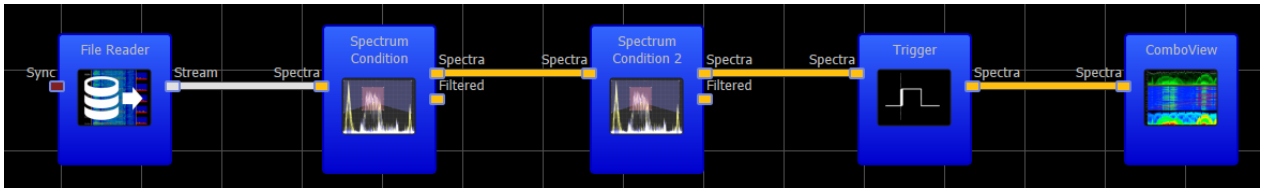
The source for this experiment Is the 2.4 GHz spectrum where we find Bluetooth together with WiFi and other signal sources.



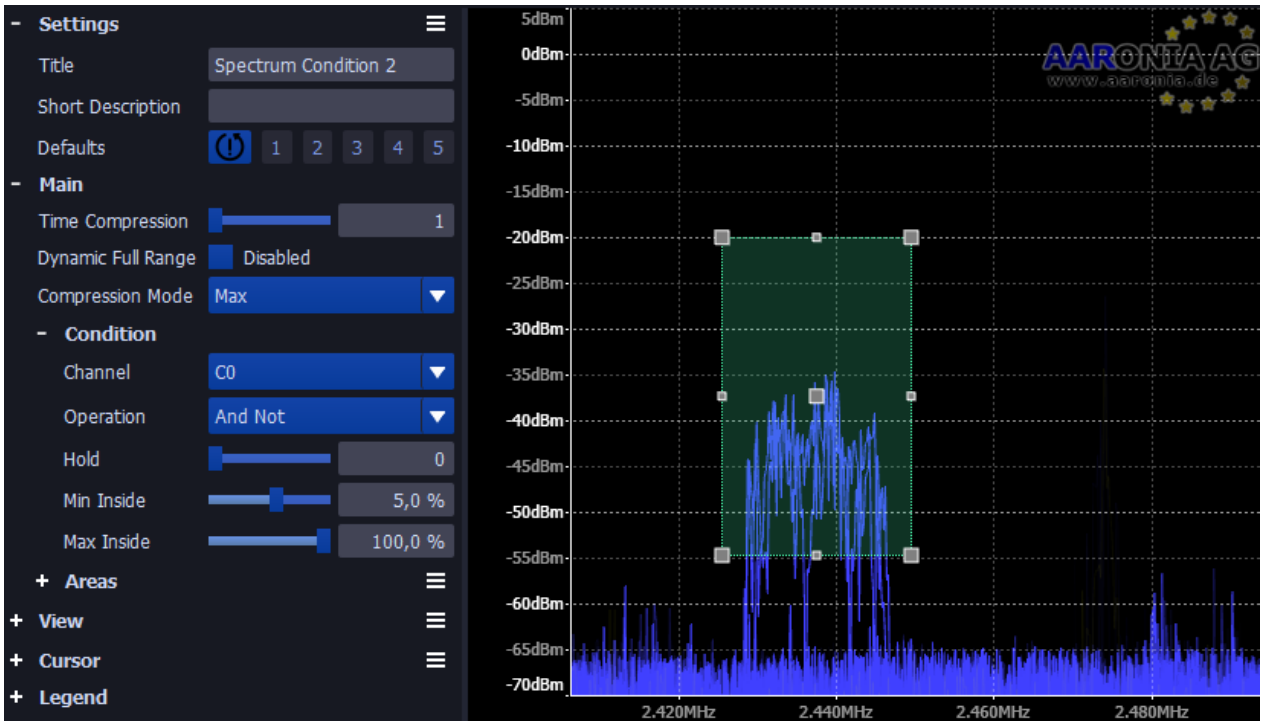
We want to extract the Bluetooth spikes to get a clearer picture of the pattern. We therefore add a condition and trigger block like the previous scenario. Unfortunately we do not only capture Bluetooth bat also a lot of the WiFi communication.



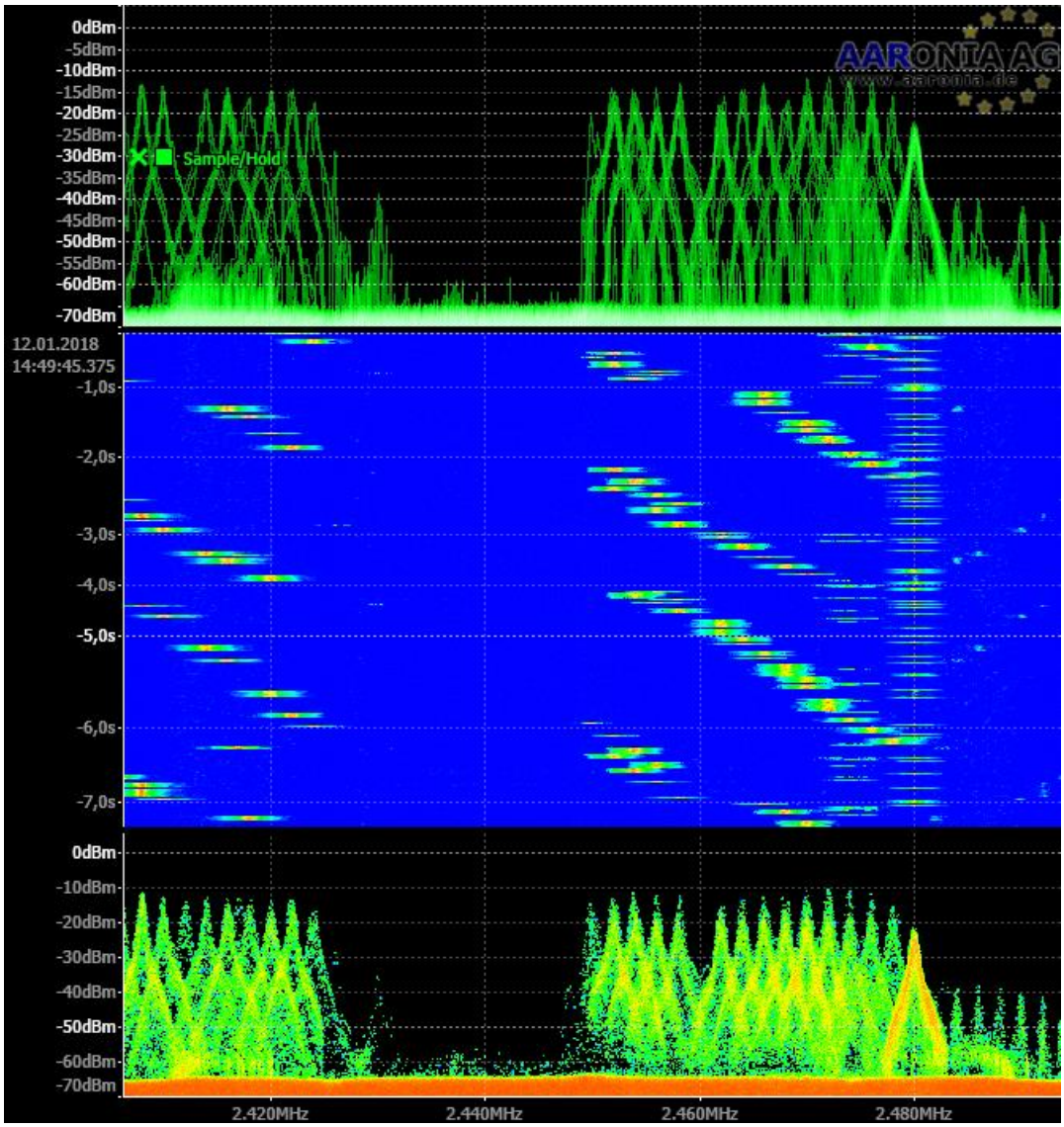
We therefore add a second condition block between the current condition block and the trigger block to remove the samples that include WiFi communication.



We mark the WiFi area as condition zone and set the minimum inside factor to 5%. We also use the “And Not” as operation to mask out all the samples that have WiFi activity.



The resulting capture now consists mainly of Bluetooth samples:


0dBm
-5dBm
-10dBm
-15dBm
-20dBm
-25dBm
-30dBm
-35dBm
-40dBm
-45dBm
-50dBm
-55dBm
-60dBm
-70dBm
Sample/Hold
AARONTA AG
12.01.2018
14:49:45.375
-1,0s
-2,0s
-3,0s
-4,0s
-5,0s
-6,0s
-7,0s
0dBm
-10dBm
-20dBm
-30dBm
-40dBm
-50dBm
-60dBm
-70dBm
2.420MHz
2.440MHz
2.460MHz
2.480MHz