Introduction

In LightningChart .NET v.10.1 a new, super-fast, line series was introduced, called **SampleDataBlockSeries**. The data is stored as memory blocks, which makes the disposing of old data, and appending new data, easier on memory and CPU.

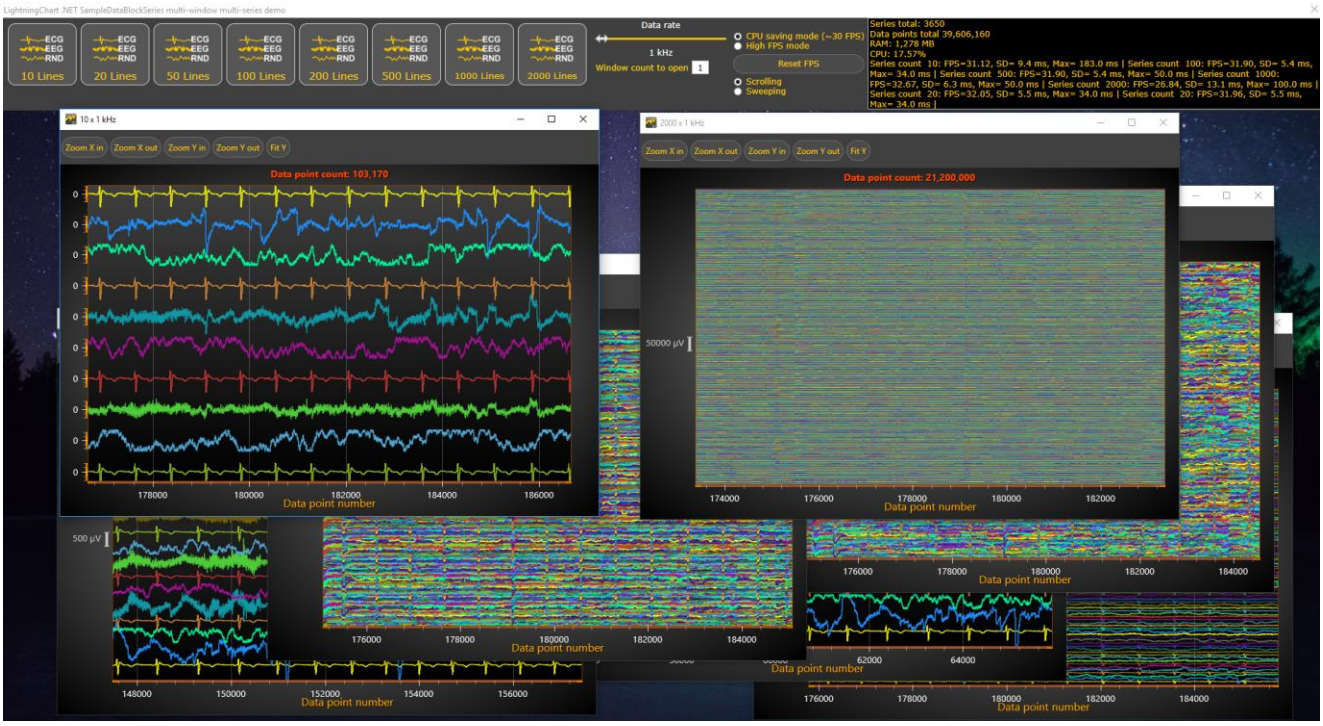
Since LightningChart v.1, there have been SampleDataSeries with linear memory array.

Both series types are made for fixed-interval data monitoring, used typically in waveform visualization, e.g., medical monitoring (ECG/EKG, EEG, EMG, ExG), vibration monitoring, telemetric and data logger systems, and audio engineering industry.

SampleDataBlockSeries rendering algorithms were designed to take better benefit from GPU computation power, freeing up more CPU resources for other processes and tasks.

Test application

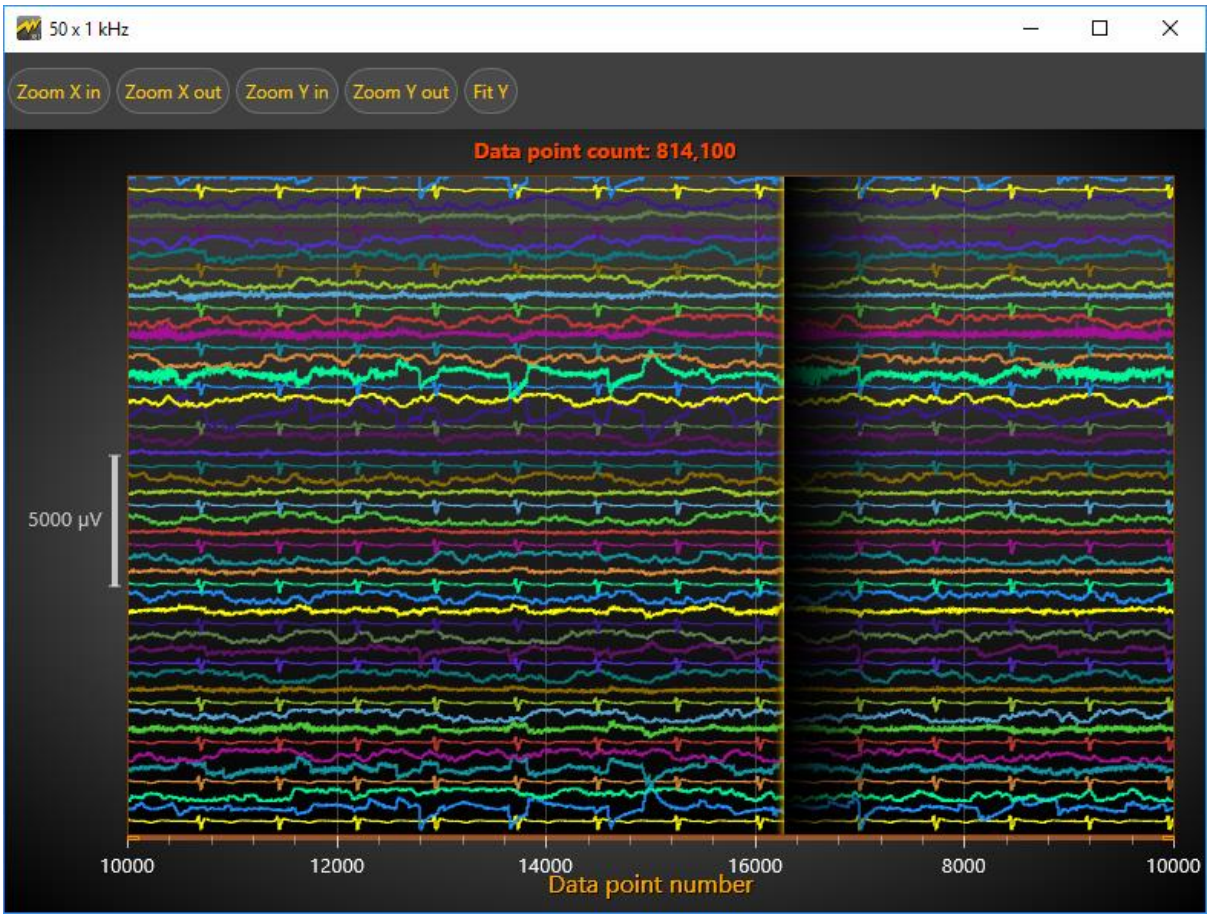
The following brief test compares these two high-performance line series in practical tests. The test was performed with a standalone ***SampleDataBlockSeries*** *demo application*.



The application consists of a header bar, and the test application allows opening dozens of chart windows. The data rate is adjustable, starting from 1000 Hz (1000 data points / sec) for each series. Each window can be opened with preferred series count, between 10... 2000.

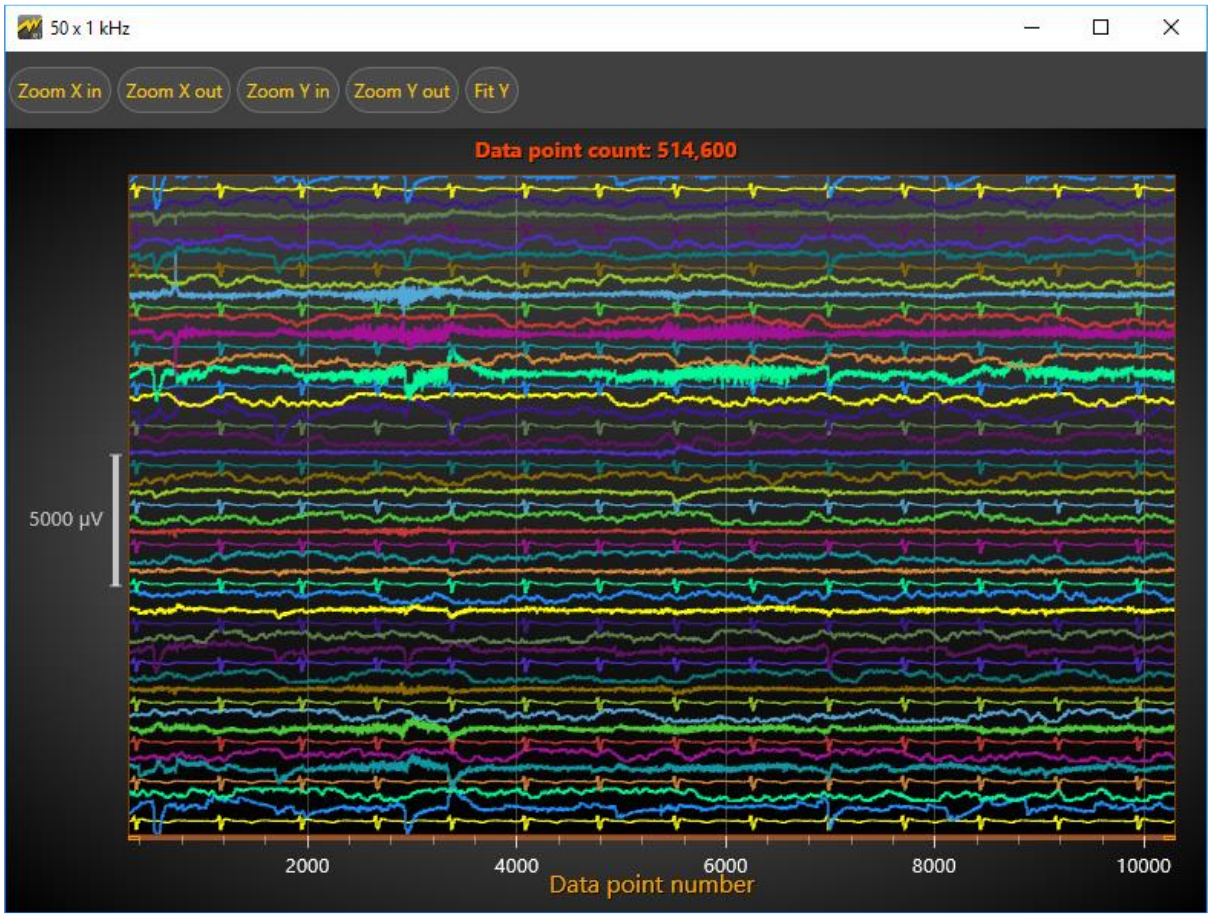
The data read from a .CSV file consisted of real ECG, EEG and then random data every 3rd series.

The chart windows can be set to the 'Sweeping' real-time scrolling mode:



Sweeping mode window

In the following image the main focus is on the **Scrolling** mode.



Scrolling mode window

The X axis length was set to hold 10 seconds of data. Line width was set to 1.5.

Performance Test Legend

The test was performed in two modes:

- CPU-saving mode
- High-FPS mode

The following criteria was measured:

- FPS (Frames / sec). A higher value is better.
- RAM consumption in MB. A lower value is better.
- CPU load in %. A lower value is better.
- Standard Deviation between refreshes in milliseconds. A lower value is better and indicates a smoother data-scrolling experience, with less twitching or stuttering.
- Maximum delay between refreshes in milliseconds. The lower the better.

Device hardware and Operating system

The computer employed was a mid-level developer's desktop PC, with AMD Ryzen 7 2700X CPU, 64 GB RAM, and Nvidia GTX 1070 graphics card, with the latest Windows 10 operating system.

Test results

The test results were collected in the next table and each cell was colored based on their meaning for the specific parameter.

Green = Good. Yellow = satisfactory. Red = struggling

Test case	SDBS CPU saving mode					SDBS High-FPS mode mode					SDS CPU saving mode					SDS High-FPS mode mode				
	FPS	RAM	CPU	SD	Max	FPS	RAM	CPU	SD	Max	FPS	RAM	CPU	SD	Max	FPS	RAM	CPU	SD	Max
1 Window x 10 series, 1 kHz Data rate	60	198	2.3	0.5	17	480	211	7.5	1.6	20	60	273	3.5	0.5	17	284	407	12.2	1.3	19
10 Windows x 10 series, 1 kHz Data rate	36	759	13.5	8.0	34	169	802	36.6	4.9	21	34	1449	19.3	6.4	34	76	1848	50.8	4.7	33
1 Window x 100 series, 1 kHz Data rate	60	265	2.6	0.5	17	348	271	7.8	2.6	18	60	391	16.1	0.5	17	88	464	19.7	2.1	31
10 Windows x 100 series, 1 kHz Data rate	36	988	13.9	8.0	34	145	1069	45.1	5.4	37	13	2488	39.4	29.7	333	14	2664	33.5	28.8	216
1 Window x 200 series, 1 kHz Data rate	60	271	3.3	0.6	33	265	283	8.3	1.3	18	55	463	21.9	6.7	50	44	521	21.5	4.7	53
10 Windows x 200 series, 1 kHz Data rate	33	1066	17.3	7.5	34	106	1190	42.6	4.8	25	5	4192	19.8	122.6	617	5	3925	35.0	127.0	770
1 Window x 1000 series, 1 kHz Data rate	60	418	7.4	2.4	34	64	454	8.2	3.8	36	11	1036	23.1	15.0	166	10	10678	20.2	14.7	218
5 Windows x 1000 series, 1 kHz Data rate	32	1143	20.8	4.7	34	54	1164	32.6	3.8	38	3	5018	35.0	182.4	1083	2.7	5298	34.0	219.0	990
10 Windows x 1000 series, 1 kHz Data rate	27	2091	41.6	8.7	67	27	2091	52.1	6.3	70	1	10216	24.1	1095.4	3634	0.53	11611	15.8	1053.0	5870
1 Window x 10 series, 10 kHz Data rate	60	261	2.6	0.5	17	455	290	8.0	1.6	18	60	1978	4.2	7.1	233	277	510	12.0	3.2	291
10 Windows x 10 series, 10 kHz Data rate	36	839	15.2	8.0	34	167	914	37.0	6.2	28	35	1972	20.9	7.1	34	75	2548	49.3	5.0	29
1 Window x 100 series, 10 kHz Data rate	60	334	3.0	0.5	17	396	406	9.3	1.4	17	60	629	12.1	0.5	33	79	835	23.1	2.5	44
10 Windows x 100 series, 10 kHz Data rate	36	1626	17.0	7.9	34	138	1779	45.2	5.2	29	12	4826	40.2	30.8	167	14	5223	43.6	32.5	236
1 Window x 200 series, 10 kHz Data rate	60	335	3.0	0.5	17	272	391	8.0	1.7	45	47	918	14.3	7.4	50	48	805	21.3	3.9	48
10 Windows x 200 series, 10 kHz Data rate	36	1849	17.1	8.2	34	92	1937	46.4	4.8	30	5	6790	21.3	128.0	700	5	7051	32.9	127.7	605
1 Window x 1000 series, 10 kHz Data rate	58	500	8.9	2.8	34	67	575	7.5	3.7	38	9	1885	19.3	31.5	250	9	1968	12.1	17.7	171
5 Windows x 1000 series, 10 kHz Data rate	36	1628	23.1	7.9	50	45	1580	28.3	4.5	52	1	8655	26.9	633.1	2500	1	8688	32.2	561.0	2072
10 Windows x 1000 series, 10 kHz Data rate	18	3083	38.9	12.7	117	17	3093	42.2	18.2	147	0.3	26494	23.2	1624.0	6100	0.4	20995	33.3	1109.0	4708

The CPU column was left without a color as its load depends on the preferred frame rate *(more about this later)*.

Analysis

By visually comparing the previous color-coded results, we can already see SampleDataBlockSeries to perform significantly better than SampleDataSeries, which has more red color all over the table.

However, by looking at the following table based on the test results, it is possible to compare how many times the SampleDataBlockSeries are better than the SampleDataSeries.

Test case	Comparison SDBS vs. SDS How many times better is SDBS?										
	RAM CPU-saving mode	**FPS** CPU-saving mode	**FPS** High-FPS mode	**SD** CPU-saving mode	**SD** High-FPS mode	**Max** CPU-saving mode	**Max** High-FPS mode	**CPU** CPU-saving mode. Note SDBS is having higher FPS	**CPU load** High-FPS mode. Note SDBS is having higher FPS	**CPU efficiency, load vs. refresh rate** CPU-saving mode	**CPU efficiency, load vs. refresh rate** High-FPS mode
1 Window x 10 series, 1 kHz Data rate	1.4	1.0	1.7	1.0	0.8	1.0	1.0	1.5	1.6	1.5	2.8
10 Windows x 10 series, 1 kHz Data rate	1.9	1.1	2.2	0.8	1.0	1.0	1.6	1.4	1.4	1.5	3.1
1 Window x 100 series, 1 kHz Data rate	1.5	1.0	4.0	1.0	0.8	1.0	1.7	6.2	2.5	6.2	10.0
10 Windows x 100 series, 1 kHz Data rate	2.5	2.8	10.4	3.7	5.3	9.8	5.8	2.8	0.7	7.9	7.7
1 Window x 200 series, 1 kHz Data rate	1.7	1.1	6.0	11.2	3.6	1.5	2.9	**6.6**	2.6	7.2	15.6
10 Windows x 200 series, 1 kHz Data rate	3.9	6.6	21.2	16.3	26.5	18.1	30.8	1.1	0.8	7.5	17.4
1 Window x 1000 series, 1 kHz Data rate	2.5	5.5	6.4	6.3	3.9	4.9	6.1	3.1	2.5	17.0	15.8
5 Windows x 1000 series, 1 kHz Data rate	4.4	10.7	20.0	38.8	57.6	31.9	26.1	1.7	1.0	18.0	20.9
10 Windows x 1000 series, 1 kHz Data rate	4.9	27.0	**50.9**	125.9	**167.1**	**54.2**	**83.9**	0.6	0.3	15.6	15.4
1 Window x 10 series, 10 kHz Data rate	7.6	1.0	1.6	14.2	2.0	13.7	16.2	1.6	1.5	1.6	2.5
10 Windows x 10 series, 10 kHz Data rate	2.4	1.0	2.2	0.9	0.8	1.0	1.0	1.4	1.3	1.4	3.0
1 Window x 100 series, 10 kHz Data rate	1.9	1.0	5.0	1.0	1.8	1.9	2.6	4.1	2.5	4.1	12.5
10 Windows x 100 series, 10 kHz Data rate	3.0	3.0	9.9	3.9	6.3	4.9	8.1	2.4	1.0	7.1	9.5
1 Window x 200 series, 10 kHz Data rate	2.7	1.3	5.7	14.8	2.3	2.9	1.1	4.7	**2.6**	6.0	15.0
10 Windows x 200 series, 10 kHz Data rate	3.7	7.2	18.4	15.6	26.6	20.6	20.2	1.2	0.7	9.0	13.1
1 Window x 1000 series, 10 kHz Data rate	3.8	6.4	7.4	11.3	4.8	7.4	4.5	2.2	1.6	13.9	12.0
5 Windows x 1000 series, 10 kHz Data rate	5.3	36.0	45.0	80.1	124.7	50.0	39.8	1.2	1.1	**42.0**	**51.2**
10 Windows x 1000 series, 10 kHz Data rate	**8.6**	**60.0**	42.5	**127.9**	60.9	52.1	32.0	0.6	0.8	35.8	33.6

- In RAM consumption, up to 8.6 times better.
- In FPS (CPU-saving mode), up to 60 times better.
- In FPS (High-FPS mode), up to 51 times better.
- In Standard deviation of refresh intervals (CPU-saving mode), up to 128 times better.
- In Standard deviation of refresh intervals (High-FPS mode), up to 167 times better.
- In longest refresh interval durations (CPU-saving mode), up to 54 times better.
- In longest refresh interval durations (High-FPS mode), up to 84 times better.

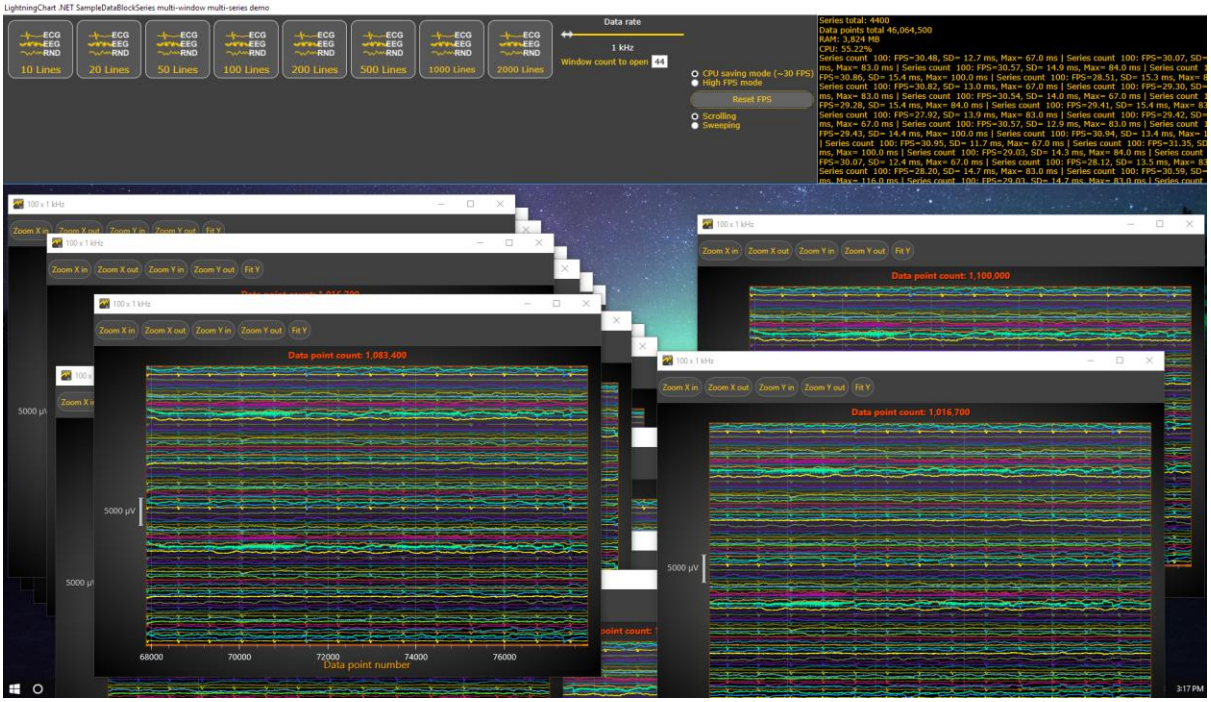
Commonly in all tests, the CPU load is lower than with the SampleDataSeries, yet keeping the refresh rate higher. Finally, we can calculate how many times more **CPU efficient** SampleDataBlockSeries is, when we relate it to FPS rate.

- In CPU-saving mode, CPU efficiency is up to 42 times better.
- In High-FPS mode, CPU efficiency is up to 51 times better.

Additional test

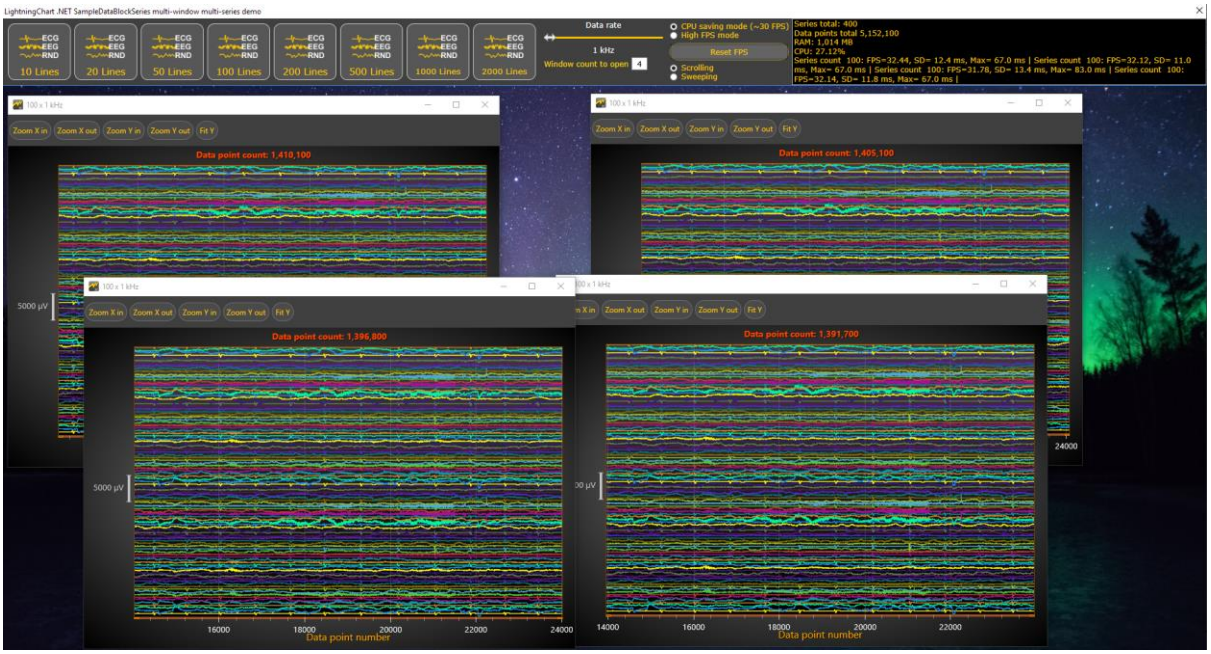
Out of curiosity, we can test how many 100-series we can open, so they still refresh around 30 FPS.

For SampleDataBlockSeries, we can open simultaneously **44** windows.



44 SampleDataBlockSeries windows

For SampleDataSeries, we can only open **4** windows simultaneously.



4 SampleDataBlockSeries windows

In conclusion, **11 times more windows** than with the SampleDataSeries.

Further Performance Information

By using a billion data points example, it is possible to visualize up to **16 billion data points** with the **SampleDataBlockSeries**, whereas the SampleDataSeries capacity is limited to **8 billion data points** with the implemented 64 GB system memory.

Conclusion

The **SampleDataBlockSeries** is the optimal series type to visualize scrolling/sweeping real-time data.

With very low variations in refresh intervals, it provides a smooth monitoring experience which is much better than SampleDataSeries.

Additionally, the **SampleDataBlockSeries** uses less RAM and CPU resources, provides a higher FPS rate and a significantly better responsiveness to user interactions.