

## ArtUs RF Data Control

The native C++ SDK sample **ArtUs RF Data Control** should be used for the beamformed RF data capture, and recording into the data files. The sample allows to control few parameters of ultrasound scanner **ArtUs** and to define dimensions of RF data stream. For the explicit description of the sample we refer to [1]. The front panel of the demo sample is presented in the Fig. below. The RF window (11) defines the amount of RF data to be transferred and the relative position of the RF output window in the image frame (10). For the convenience in the user manual RF window data and the main acquisition parameters sometimes will be called sub-frame. The sample allows to record RF window data to the file (6). The front panel and main controls of ArtUs RF Data Control sample are presented below:

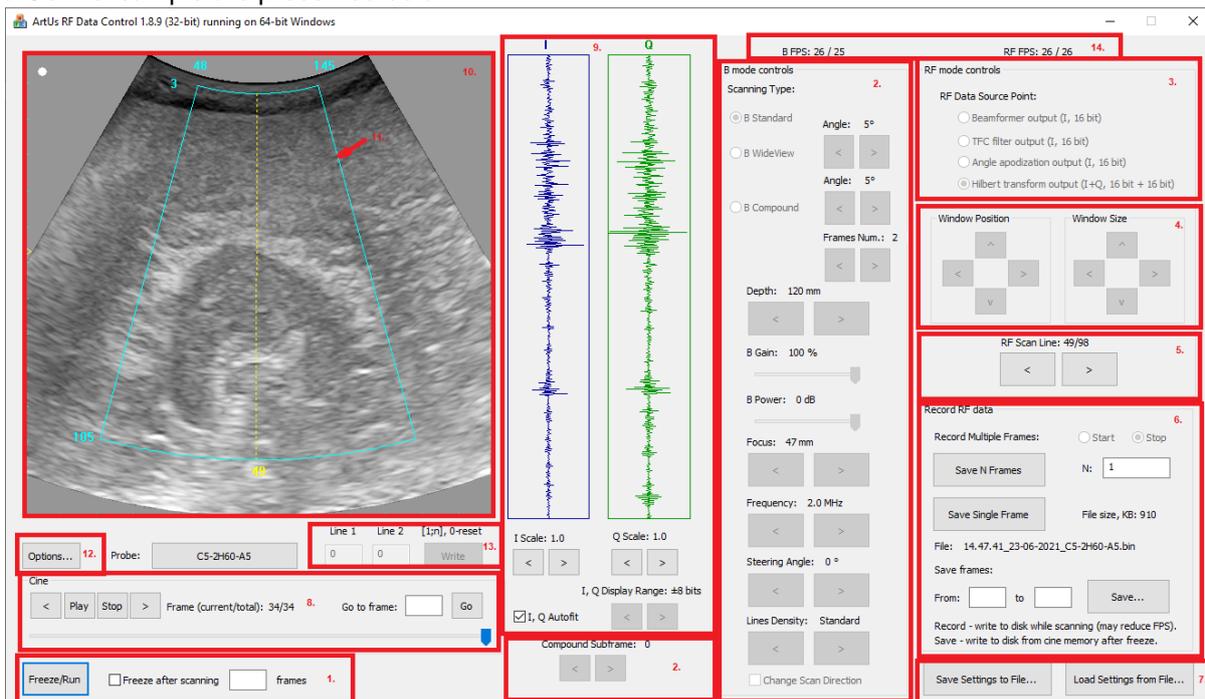


Fig. Front panel of ArtUs RF Data Control demo

Num	Component	Description
1	<b>Main scanning controls</b>	Controls starts ( <b>Run</b> ) and stops ( <b>Freeze</b> ) ultrasound scanning.
2	<b>B mode controls</b>	Allows to select scanning type ( <b>B Standard</b> , <b>B WideView</b> , <b>B Compound</b> ), to adjust scanning depth ( <b>Depth</b> , mm), amplification of the image data ( <b>B Gain</b> , %), power of transmitted ultrasound signal ( <b>B Power</b> , dB), focal depth ( <b>Focus</b> ), scanning frequency ( <b>Frequency</b> ), steering angle ( <b>Steering Angle</b> ) and lines density ( <b>Lines Density</b> ). There is an option to select scan direction ( <b>Change Scan Direction</b> ).
3	<b>RF Data Source Point</b>	RF data is received as an array of 16-bit integers. This option allows to select the point of the beamformer from which RF data will be received. For ArtUs ultrasound scanner the following RF data types are available: <ul style="list-style-type: none"> <li>• Beamformer output (<i>I</i>, 16 bit),</li> <li>• TFC (time-frequency control) filter output (<i>I</i>, 16 bit),</li> <li>• Angle apodization output (<i>I</i>, 16 bit),</li> <li>• Hilbert transform output (<i>I+Q</i>, 16 bit + 16 bit).</li> </ul> For the further details related to these data types we refer to [1].
4	<b>RF window Position and Size</b>	The position and size of RF window could be adjusted by the controls. There is a trade-off between RF window size and scanning speed (fps).

5	<b>RF Scan Line</b>	These controls allow to select RF scanning line which will be displayed on screen.
6	<b>Record RF data</b>	<p>RF data recording into (*.bin) file from the selected data source point and RF window (11). Preselected number of frames (N) is recorded into the file (default value – 100). The position and size of RF window can be adjusted by the controls. The file size could be controlled by observing indications of the counter (size in MB). There is an option to record single RF frame (button <b>Record Single Frame</b>) which could be selected from cine loop. The RF frames could be recorded by pressing Start/Stop radio buttons as well. Recording of RF data while scanning may result into reduced FPS (depends on the disk).</p> <p><b>Note! There is a small number of randomly skipped frames in RF data records when the file is recorded directly to disc in real-time during scanning. It is recommended to save RF data from Cine buffer after scanning when freeze button is pressed. Such writing ensures that all collected RF data will be recorded without missing frames and reduction of FPS.</b></p>
7	<b>Save/Load settings</b>	Allows to save current scanning and RF window settings to file, which could be loaded and settings reset.
8	<b>Cine</b>	Cine loop options. Allows to play recorded sequence and to watch it frame by frame.
9	<b>RF data plot</b>	Displays the RF data of the single selected line (dashed yellow line superimposed on the B mode image marks the scanline position). Controls (I Scale, Q Scale) allows to adjust the scale of shown RF signal. The displayed RF signals are scaled by both axes in order to obtain well visible waveforms.
10	<b>Main B mode imaging window</b>	Displays full grayscale B mode images and the superimposed boundary lines of adjustable RF window.
11	<b>RF window</b>	The contour (solid light blue lines) outlines the external boundaries of the RF window. The scanning line numbers and imaging depth are displayed close to the border.
12	<b>Options</b>	Options allows to adjust size of acquired Cine in MB. Maximum allowed size 1200 MB. Takes effect after software restart.
13	<b>Two scanning lines mode</b>	This mode scans only two selected (Line1, Line2) scanning lines and allows to achieve much higher FPS.
14	<b>FPS</b>	<p>There are FPS indicators <b>B FPS: X / Y</b> and <b>RF FPS: X / Y</b> where:</p> <p>X – screen rendering speed, depends on PC performance          Y – true scanning speed</p>

### Structure of the RF data file

The collected RF data and the main acquisition parameters needed for off-line analysis and imaging could be recorded into binary files (\*.bin). The filename contains acquisition time and date information, and the probe type code (**HH.MM.SS\_DD-MM-YYYY\_probe\_code.bin**, i.e. "16.34.00\_27-10-2017\_L18-10H30-A4.bin"). The **ArtUs** RF Data Control SDK allows to record the RF data files of unlimited size (limited only by the capacity of the HDD).

Each recorded RF data sub-frame contains header information and beamformed RF data from the corresponding RF window. The position and size of the window could be adjusted during scanning, and therefore the header of acquisition information is written to file for each sub-frame in a recorded sequence. Please note that smaller window size allows to achieve higher frame rates. The file structure for single sub-frame is defined as follows:

Contents of the file	Data type	Number of elements (bytes)	Description
<i>RF file type</i>	char	1 (6 bytes)	RF file version is written only in the beginning of the file not before each frame (currently <b>RF0004</b> ).
<i>number_of_frames</i>	int32	1 (4 bytes)	Number of RF frames recorded into file ( <b>Note:</b> if Record Multiple Frames Start/Stop option is used during recording of RF data the field will be equal to 0).
<i>header_size</i>	int32	1 (4 bytes)	Size of header in bytes
<i>frame_size</i>	int32	1 (4 bytes)	Size of RF frame in bytes
<i>source_ID</i>	int32	1 (4 bytes)	RF data source ID: 1 – Beamformer output, 2 – TFC filter output, 3 – Angle apodization output, 4 – Hilbert transform output
<i>tx_frequency</i>	int32	1 (4 bytes)	US wave transmission frequency, in Hz.
<i>frame_rate</i>	int32	1 (4 bytes)	The actual number of frames per second, since streaming started, depends on the used scanning probe and size of RF window. Frame rate is multiplied by 100.
<i>Length_of_RF_row</i>	int32	1 (4 bytes)	Number of RF signal samples in a single scanning row.
<i>Number_of_RF_rows</i>	int32	1 (4 bytes)	Number of RF scanning lines in a predefined RF window.
<i>Sampling_period_ns</i>	int32	1 (4 bytes)	Sampling period – defines the digitization resolution, in nanoseconds (ns). For the ArtUs system - 25 ns ~ 40 MHz sampling for depths up to 8192 samples, and 50 ns (20 MHz) for number of samples >8192.
<i>Sample_size</i>	int32	1 (4 bytes)	Number of BIT's, at present 16/32 BIT for the RF data
<i>start_depth</i>	int32	1 (4 bytes)	Scanning start depth in mm (offset for each beam) used for imaging of the scanning window in absolute coordinate system
<i>Start_point_position_and_orientation</i>	int32	3 × Number_of_RF_rows (3 × Number_of_RF_rows × 4 bytes)	<p>Start position coordinates and orientation angle of each beam are originally recorded in triplets (beam_x<sub>i</sub>, beam_y<sub>i</sub>, angle<sub>i</sub>, where <i>i</i>-th scanning beam) for each scanning beam beginning from left side of the sector.</p> <ul style="list-style-type: none"> <li>• angle – specifies the angle of the ultrasonic beam's direction (in radians multiplied by 1000000). The angle is given relative to the perpendicular to the center of the probe's surface (angle = 0 for linear array probes); in case of compound scanning mode (B compound) the parameter returns angle of beams for each compounded sub-frame (number of compounded sub-frames could be adjusted by Frames Num. control);</li> <li>• beam_x - start x coordinates of each beam, (in μm);</li> <li>• beam_y - start y coordinates of each beam, (in μm).</li> </ul>
<i>time_stamps</i>	uint32	Number_of_RF_rows (Number_of_RF_rows × 4 bytes)	Time stamps for each ultrasound scanning line. Timestamp – time from the last start of scanning until the start of receiving of RF-data in present ultrasound line. One discrete of Timestamp = sampling period (25 ns for 40 MHz sampling clock of ArtUs).

Maximum time for 32-bit timestamp is 28606043.625  $\mu$ sec (28.605043625 sec), after that timestamp counter begins count from zero. Timestamps are in samples.

<i>frame</i>	<i>int16</i>	$\text{Number\_of\_RF\_rows} \times \text{Length\_of\_RF\_row}$ $(\text{Number\_of\_RF\_rows} \times \text{Length\_of\_RF\_row} \times 2 \text{ bytes})$ for <i>source_ID</i> =1, 2, 3; <i>and</i> $2 \times \text{Number\_of\_RF\_rows} \times \text{Length\_of\_RF\_row}$ $(2 \times \text{Number\_of\_RF\_rows} \times \text{Length\_of\_RF\_row} \times 2 \text{ bytes})$ for <i>source_ID</i> =4	Single sub-frame RF data in 1D vector format (int16);
--------------	--------------	---	---

### Structure of the file:

#### Source\_ID=1, 2, 3

4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	3 × Number_of_RF_rows × 4 bytes	Number_of_RF_rows × 4 bytes	Number_of_RF_rows × Length_of_RF_row × 2 bytes	
<i>number_of_frames</i>	<i>header_size</i>	<i>frame_size</i>	<i>source_ID</i>	<i>tx_freq</i>	<i>frame_rate</i>	<i>Length_of_RF_row</i>	<i>Number_of_RF_rows</i>	<i>Sampling_period_ns</i>	<i>Sample_size</i>	<i>start_depth</i>	<i>Start_point_position_and_orientation</i>	<i>time_stamps</i>		<i>frame</i>	

#### Source\_ID=4 (Hilbert transform output)

4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	3 × Number_of_RF_rows × 4 bytes	Number_of_RF_rows × 4 bytes	2 × Number_of_RF_rows × Length_of_RF_row × 2 bytes	
<i>number_of_frames</i>	<i>header_size</i>	<i>frame_size</i>	<i>source_ID</i>	<i>tx_freq</i>	<i>frame_rate</i>	<i>Length_of_RF_row</i>	<i>Number_of_RF_rows</i>	<i>Sampling_period_ns</i>	<i>Sample_size</i>	<i>start_depth</i>	<i>Start_point_position_and_orientation</i>	<i>time_stamps</i>	<i>I component</i>	<i>Q component</i>	

### Known Issues

- There is a small number of randomly skipped frames in RF data records when the file is recorded directly to disc in real-time during scanning. The recording itself during scanning may reduce FPS (depends on disc speed). If the exact moments of time are important, we recommend to check time stamps in RF data recordings. Overall, the missing frames are caused due to unpredictable behaviour of windows task scheduler which distributes resources for different processes such as B mode stream, RF data stream, writing to disc etc. **We recommend to record RF data files after scanning from the Cine loop memory.** There will be all frames without losses. The same proposal is valid for the other developments, since you can use SDK and retrieve RF data from Cine loop memory.
- Artus\_rf\_data\_control tool indicates number of rendered frames per second. USGFW2 SDK architecture assumes that ultrasound data captured from beamformer hardware is stored into cineloop buffer with frame rate ultrasound scanning is performed. But rendering is performed with FPS that Windows OS allows to process. If system does not allow to render all frames stored in cineloop buffer because of ultrasound data processing calculations or because of Windows threads manager switch algorithm some frame in cineloop is skipped to save ultrasound streaming timings. It is a reason why you can see FPS in artus\_rf\_control less than real scanning FPS. You can also observe different FPS for B and RF data rendering because Windows threads switch manager decided to dedicate more time to one of (B or RF) processing threads.

### References

[1] Echo Blaster 64, Echo Blaster 128, LS64, LS128, ClarUs, SmartUs, MicrUs and ArtUs Series Ultrasound Systems Ultrasonography for Windows II Software Development Kit (Usgfw2 SDK) Programmer's Guide, (main sub-section – 4.9).

**Revision History**

<b>Version</b>	<b>Date</b>	<b>Description of Revision</b>	<b>Revision author</b>
1.0.0	02/09/2019	Initial Release	A. Sakalauskas
1.1.0	10/09/2019	Added option to record single RF frame	A. Sakalauskas
1.2.0	30/12/2019	Added option to record predefined number of RF frames and save file dialog window. Structure of the file was modified.	A. Sakalauskas
1.3.0	22/04/2020	Added option to record I and Q components into RF data file for Hilbert transform output; Changed display of RF window line numbers to start from 1; RF window movement step changed from 2 to 1.	A. Sakalauskas
1.4.0	05/05/2020	Added option to record time stamps for each scanning line into RF data file	A. Sakalauskas
1.4.1	10/07/2020	Added option to adjust Cine buffer size, two scanning lines mode.	V.Perlibakas A.Sakalauskas
1.4.2	18/01/2021	Fixed writing of beam coordinates and angles to RF file in compound mode.	A. Portniagin V. Perlibakas
1.4.3	12/04/2021	Added section "Known issues"	A. Sakalauskas
1.4.4	19/04/2021	B and RF mode FPS output changed from "Drawing FPS" to "Drawing FPS / Scanning FPS"	V.Perlibakas
1.4.5	23/06/2021	Sampling period for higher depths (>8192 samples) automatically set to 50 ns.	A. Portniagin A. Prysiazniuk
1.4.6	11/04/2022	Fixed bug of writing number of frames field into a RF data file, RF file structure version RF0004 created.	V.Perlibakas A.Sakalauskas