SeeSharpTools

User Manual

Version 1.5.0

Contents

1. SeeSharpTools Summary 4
   1.1 SeeSharpTools Introduction 4
   1.2 SeeSharpTools Q&A 5
2. SeeSharpTools Class Library Summary 6
   2.1 SeeSharpTools.JY.ArrayUtility Class Library 6
   2.2 SeeSharpTools.JY.DSP.Fundamental Class Library 8
   2.3 SeeSharpTools.JY.DSP.Measurement Class Library 11
   2.4 SeeSharpTools.JY.GUI Class Library 15
      2.4.1 EasyChart 15
      2.4.2 AquaGauge 17
      2.4.3 PressureGauge 18
      2.4.4 Thermometer 18
      2.4.5 KnobControl 19
      2.4.6 Slide 20
      2.4.7 Led 22
      2.4.8 EasyButton 22
      2.4.9 IndustrySwitch 23
      2.4.10 ButtonSwitch 24
      2.4.11 Tank 24
      2.4.12 ScrollingText 25
      2.4.13 SevenSegment 26
      2.4.14 SegmentBright 26
      2.4.15 LedMatrixControl 27
      2.4.16 SwitchArray and LEDArray 28
      2.4.17 JYArray 28
      2.4.18 PathControl 29
      2.4.19 StripChart 30
      2.4.20 GaugeLinear 32
      2.4.21 EasyChartX 32
      2.4.22 ViewController 39
      2.4.23 StripChartX 40
   2.5 SeeSharpTools.JY.Database Class Library 45
   2.6 SeeSharpTools.JY.Localization Class Library 46

2.7 SeeSharpTools.JY.DSP.FilterMCR Class Library 46
2.8 SeeSharpTools.JY.DSP.SoundVibration Class Library 47
2.9 SeeSharpTools.JY.DSP.Utility Class Library 48
2.10 SeeSharpTools.JY.Graph3D Class Library 52
2.10.1 Surface Graph Class 52
2.10.2 Intensity Graph Class 53
2.11 SeeSharpTools.JY.File Class Library 55
2.11.1 AnalogWaveformFile Class 55
2.11.2 BinHandler Class 57
2.11.3 CsvHandler Class 57
2.11.4 IniHandler Class 58
2.12 SeeSharpTools.JY.Report Class Library 60
2.12.1 WordReport Class 60
2.12.2 ExcelReport Class 60
2.12.3 Logger Class 61
2.13 SeeSharpTools.JY.ThreadSafeQueue Class Library 63
2.13.1 ThreadSafeQueue Class 63
2.13.2 CircularQueueClass 63
2.14 SeeSharpTools.JY.Sensors Class Library 64
2.14.1 CustomScaling Class 64
2.14.2 DisplacementSensor Class 64
2.14.3 LoadCell Class 64
2.14.4 RTD Class 65
2.14.5 Thermistor Class 65
2.14.6 Thermocouple Class 65
2.15 SeeSharpTools.JY.Statistics Class Library 66
2.16 SeeSharpTools.JY.TCP Class Library 67
2.17 SeeSharpTools.JY.Mathemetics Class Library 69
3. FAQ 75

1. SeeSharpTools Summary

1.1 SeeSharpTools Introduction

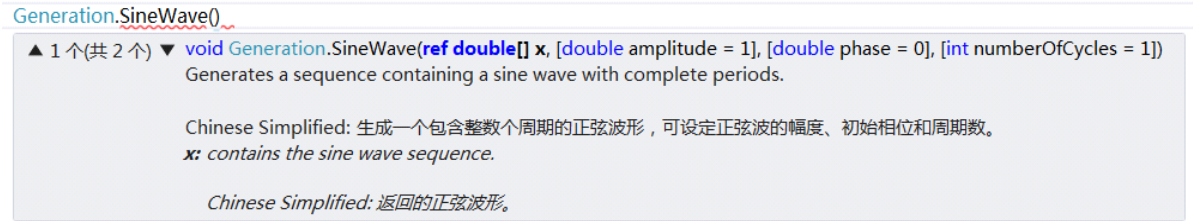
SeeSharpTools software is a free software component provided byJYTEK SeeSharp Platform. It includes a series of Class Libraries, providing easy-to-use signal generation, analysis and display functions to help you quickly build test and measurements solutions under the SeeSharp Platform. The description of the libraries in SeeSharpTools software package is listed in the following table:

Class Library Name	Function Summary
JY.ArrayUtility	Provides common array operations and operational arithmetic functions.
JY.Audio	Provides waveform generation functions and metric algorithms commonly used in audio testing
JY.DSP.Fundamental	Provides common waveform generation and spectrum calculation functions
JY.GUI	Provide common controls and related functions for test and measurement
JY.Database	Provide common functions of the database
JY.Localization	Provide localization for WinForms programs
JY.DSP.FilterMCR	Provide Matlab-based filter calculation related functions
JY.DSP.SoundVibration	Provides algorithmic functions for sound vibration
JY.DSP.Utility	Provide common data processing algorithm functions
JY.DSP.Utility.Fundmental	Provide common basic digital signal processing algorithms
JY.Graph3D	Provide 3D graphics functions
JY.File	Provide file reading and writing
JY.Report	Provide reporting functions
JY.ThreadSafeQueue	Provide a set of thread-safe queue components
JY.Sensors	Providing a set of data conversion components of the sensor
JY.Statistics	Algorithmic tool for providing statistical calculations
JY.TCP	Provide a set of functional interfaces for transmitting data over a network socket

1.2 SeeSharpTools Q&A

Q: Does SeeSharpTools software provide online help documentation in the development environment?

A: Yes, when you call the methods provided by the various libraries of SeeSharpTools in the development environment, the online help is automatically displayed (as shown below). The detailed content of the various classes of SeeSharpTools is also covered in the rest of this document.



Q: Does the SeeSharpTools software include more common functions such as digital signal processing and file reading and writing?

A: Under the SeeSharp measurement and control platform, the company provides a richer set of test and measurement related functions in the form of routine source code. For details, please refer to the SeeSharpExamples routine set. In addition, under the sharp measurement and control platform, there are many free third-party free or open source libraries. For the use of these third-party libraries, the company is also provided in the form of routines and reference designs.

2. SeeSharpTools Class Library Summary

The SeeSharpTools class library provides a set of commonly used components for test and measurement, covering interface design controls, array operations, signal processing, file reading and writing, database operations, etc..

2.1 SeeSharpTools.JY.ArrayUtility Class Library

JY.ArrayUtility contains two static classes: ArrayCalculation and ArrayManipulation, which provide common array calculation and operation functions. The methods provided by these classes are all template methods and the following data type is supported : double, float, decimal, long, ulong, int, uint, short, ushort, char, byte。.The name and description of all the methods are listed in the following table:

ArrayCalculation Class	
Method Name	**Function Summary**
InitializeArray	Initialize array with given value.
Add Subtract Multiply	Addition, subtraction, and multiplication of elements one by one between two equal-length one-dimensional arrays of values.
AddOffset SubtractOffset MultiplyScale	Addition, subtraction, and multiplication of a one- or two-dimensional array of values and a single value.
Zero	Sets the value of all elements in a one- or two-dimensional array of values to zero.
Sum Average RMS Abs	The calculation of the sum, mean, root mean square, and absolute values of a one-dimensional array of values.
AreEqual	Compare two numeric arrays to be identical.
ArrayMaxAndMin	Find the maximum and minimum values and their indexes in a one-dimensional array; Find the maximum and minimum values and their indexes in a two-

	dimensional array.

ArrayManipulation Class	
Method Name	**Function Summary**
BuildArray	Concatenate two two-dimensional arrays, and the elements beyond the dimension will be discarded; Concatenate two two-dimensional arrays, and the elements beyond the dimension will be discarded.
Insert_1D_Array	Insert element to one-demensional array in specified position.
Insert_2D_Array	Insert array to specified column of two dimensional array
Connect_1D_Array	Connect two one-dimensional array to a one-dimensional array
Connected_2D_Array	Connect two two-dimensional array by row.
Concatenate	Connect two or more arrayes (one dimensional or two dimensional) to a two dimensional array by row or by column.
Convert2StringArray	Convert array data to a same-sized string array, the formatting style can be configured.
Convert2NumericArray	Convert string array data to a same-sized numeric array.
GetArraySubset	Copy subsequent elements from the specified starting position in the one-dimensional array a[] to the one-dimensional array b[]; Copy the specified row or column of the two-dimensional array a[,] to the one-dimensional array b[]; Get the specified row or column of the two-dimensional array src[,] and return it to the one-dimensional array. The row (or column) index value is given by index, the number of rows or columns is given by count, and the index by row or column is given by indextype.
ReplaceArraySubset	Copy all elements of the one-dimensional array a[] to the specified position in the one-dimensional array b[], replacing the original elements; Copy all elements of the one-dimensional array a[] to the specified row or column in the two-dimensional array b[,], replacing the elements of the row or column; Copy all elements of the two-dimensional array src[,] to the specified position in the two-dimensional array dst[,], replace the original elements, and the value of the starting index row/column is given by rowindex/columnindex.
InsertArraySubset	Insert a new two-dimensional array insertarray [,] at the specified index position of the original two-dimensional array SRC [,], the position of the inserted row or column is given by index, and the direction of the inserted row and column is given by indextype.
SplitArray	Split a one-dimensional array at the specified index; Split a tow-dimensional array at the specified index.

Interleave_1D_Arrays	Interleave the corresponding elements in a plurality of one-dimensional arrays to form a new output array.When the size of the input array is different, the number of elements of the interleaved array is equal to the minimum number of elements of input array multiplied by the number of input arrays.
Decimate_1D_Array	Divide the elements of the one-dimensional array into several arrays, output the arrays in turn, and discard all the elements that lead to different lengths of the output arrays.
Rotate_1D_Array	Make elements in an array move some positions.
Search_1D_Array	Search for elements from the start index in a one-dimensional array, and return the first matching index. If it is not found, return - 1.
Interpolate_1D_Array	Linearly interpolate a value at the index of the input one-dimensional array.
ThresholdInterpolate_1D_Array	Interpolate a point in a one-dimensional array and compare the threshold value with the value after the start index position until a pair of continuous elements are found.The threshold value is greater than or equal to the first element and less than or equal to the second element.
Transpose	The two-dimensional array row and column transpose, that is, the array of N rows and M columns is converted into M rows and N columns.
Reverse	Reverse the elements in a one-dimensional array. The source array and target array can be the same.
Sort	Sort the element in an one-dimensional array or two dimensional array (entire sorting or single row, single column sorting are supported) with specified direction(Ascend or Descend). The source array and target array can be the same.
Remove	Remove the specified elements in an one-dimensional array (start index and length can be configured) or two dimensional array (MajorOrder, start index and length can be configured).

2.2 SeeSharpTools.JY.DSP.Fundamental Class Library

JY.DSP.Fundamental contains two static classes: Generation and Spectrum, which provide common waveform generation and spectrum calculation functions, as shown in the following table:

Method Name	Function Summary

Generation Class	
SineWave	Generate a sinusoidal waveform containing an integer number of periods, which can set the amplitude, initial phase and number of cycles of the sine wave; Generate a sinusoidal waveform that sets the amplitude, initial phase, frequency, and sample rate of the sine wave.
SquareWave	Generate a square wave waveform containing an integer number of cycles, which can set the amplitude, duty cycle and number of cycles of the square wave; Generate a square wave waveform that sets the square wave's amplitude, duty cycle, frequency, and sample rate.
UniformWhiteNoise	Generate a random white noise waveform of the specified amplitude.
Ramp	Generate an arithmetic progression column.
Generation.Analog Class	
GaussianModulatedSinePattern	Generating sine wave with Gaussian modulation
GaussianMonopulse	Generate waveform containing Gaussian monopulse.
SincPattern	Generate sinc waveform.
PeriodicSincPattern	Generate periodic sinc waveform.
PulsePattern	Generate pulse waveform.
ChirpPattern	Generate chirp waveform.
ImpulsePattern	Generate impulse waveform.
Generation.Digital Class	
Gradient	Generate digital signal of that period is halved in turn.
Marching	Generate digital signal of that the binary value of the first signal of the first sampling will be moved to the next signal of the subsequent sampling set of the waveform in order.
Periodic	Generate multi cycle digital signal, each cycle consists of a digital signal and idle state.
Single	Generate a digital signal with all bits set to 0 or 1.
Random	Randomly generate digital signals containing 0 and 1. There is no mathematical law for the numerical generation of random waveform.
Switch	Generate digital signal of that the value 1 is used to generate binary values in even samples and the value 2 for odd samples.
Spectrum Class	
PowerSpectrum	Calculate the power spectrum of the input time domain signal, and set the unit of windowing type and output power spectrum;
AdvancedComplexFFT	Calculate the amplitude spectrum and phase spectrum of the input time domain signal. The spectrum data is returned in an array of complex data and the minimum

	length of the complex data array can be calculated by the following expression: InputDataLength/2 + 1.
PeakSpectrumAnalysis	Analysis the spectrum peak and peak frequency.
DBFullScaleSpectrum	Computes the dB full scale spectrum of input time-domain signal.

Note: Before using the Spectrum class, you need to install JXDSPRuntimeMKL. JXDSPRuntimeMKL is divided into X64 and X86 versions. Please select the installation according to the windows system.

2.3 SeeSharpTools.JY.DSP.Measurements Class Library

JY.DSP.Measurements contains 5 static classes: SquarewaveMeasurements JitterAnalysis.ClockRecovery, JitterAnalysis.Jitter, JitterAnalysis.Level and JitterAnalysis. Timing, which provide squarewaveform measurements and jitter analysis functions, as shown in the following table:

Method Name	Function Summary
SquarewaveMeasurements Class	
AmplitudeLevelAnalysis	According to the setting of high and low levels, analyze the amplitude and the level of high/low levels of a single waveform; According to the setting of high and low levels, analyze the amplitude and the level of high/low levels of multiple waveform.
PeriodAnalysis	Analyze the period, duty cycle and pulse width of a single waveform according to the settings of high and low level, polarity and reference level; Analyze the period, duty cycle and pulse width of multiple waveform according to the settings of high and low level, polarity and reference level;
PulseMeasurement	Measure the period, pulse width and duty cycle of the specified pulse of waveform according to the settings of high and low level, polarity and reference level; Measure the period, pulse width and duty cycle of the specified pulse of each waveform according to the settings of high and low level, polarity and reference level.
CycleRmsMeanAnalysis	Measure the RMS and mean value of the specified period of each waveform according to the high and low level, polarity and reference level settings; Measure the RMS and mean value of all period for each waveform according to the high and low level, polarity and reference level settings; Measure the RMS and mean value of all period according to the high and low level, polarity and reference level settings; Measure the RMS and mean value of specified period according to the high and low level, polarity and reference level settings.
TransitionMeasurement	According to the high and low level, polarity and reference level settings, measure the slope , rise and fall time and other transition information of the specified

	edge for each waveform; According to the high and low level, polarity and reference level settings, measure the slope , rise and fall time and other transition information of the specified edge.
JitterAnalysis.ClockRecovery	
MeanClock	Recover the reference clock signal from the input signal.
ConvertBitCountToBitstream	Convert the bit number information of each transition in the signal into a bit stream sequence per unit time.
CreateReferenceBitWaveform	Create an ideal reference clock waveform synchronized with the input signal.
JitterAnalysis.Level	
MeasureStateLevel	Measure the high and low level and amplitude of the input signal.
MeasureReferenceLevel	Measure the reference level according to the input signal.
FindTransitions	Measure the boundary and direction of each transition in the signal according to the set reference level.
SelectEdge	Select the corresponding edge according to the transition direction.
OvershootUndershoot	Measure the index and amplitude of overshoot and undershoot.
MeasureMaxMinVoltage	Measure the maximum and minimum voltage at each transition in the signal.
MeasurePeakToPeakVoltage	Measure the peak to peak value of each cycle in the signal.
MeasureArea	Measure the area and average value of each cycle of the signal.
MeasureRMSVoltage	Measure the voltage RMS of each cycle in the signal.
JitterAnalysis.Timing	
FindLevelCrossings	Measure the index of crossing point for the specified voltage.
MeasureRiseTime	Measure the rise time of signal transition.
MeasureFallTime	Measure the fall time of signal transition.
MeasurePeriod	Measure the period for signal with uniform transition distribution.
MeasureDutyCycle	Measure the duty cycle of the signal.
MeasurePositiveWidth	Measure the pulse duration of the rising transition (positive pulse width or high level).
MeasureNegativeWidth	Measure the pulse duration of the falling transition (negative pulse width or low level).
MeasureBurstWidth	The burst signal starts at a certain edge and passes through a certain threshold level, ends at the next edge and passes through this threshold level again, and lasts for more than the specified time.The burst signal starts at a certain edge and passes through a certain threshold level, ends at the next edge and passes

	through this threshold level again, and lasts for more than the specified time.
MeasureSetupHoldTime	Measure the establishment time and holding time of the signal.
ChannelSkew	Measure the time delay (skewness) between two signals.
MeasurePhase	Measure the phase difference of each cycle between two signals.
JitterAnalysis.Jitter	
MeasureTimeIntervalError	Measure the time difference between the level crossing position of the original signal and the level crossing position of the corresponding reference clock signal.
MeasurePeriodJitter	Measure the difference between each cycle of the original signal and the corresponding reference clock signal.
MeasuereDutyCycleDistortion	Measure signal the duty cycle distortion that represents the average difference between the positive and negative pulse widths of the signal.
MeasureCycleToCycle	Measure the difference between adjacent elements of the sequence.
MeasurePhaseNoise	Measure the signal phase noise of the jitter that is the sum of RMS of adjacent frequency components in the frequency domain.
SignalOperation	
AnalogToDigital	Convert analog signals into digital signals.
DigitalToAnalog	Convert analog signals into digital signals.
SignalOperation.Filtering	
MedianFilter	Median filter the input sequence.
FIRFilter	Filter the input sequence using a direct FIR filter specified by the fir coefficient.
IIRFilter	Filter the input sequence x by a direct type IIR filter specified by the reverse coefficient and the forward coefficient.
ZeroPhaseFilter	Filter the input sequence using zero phase filter without boundary extension.
SignalOperation.Analog	
Invert	Calculate the opposite number for each element in the array.
Convolution	Calculate the convolution of two one-dimensional input sequences using direct method of linear convolution.
AutoCorrelation	Calculate the autocorrelation of one-dimensional sequence.
CrossCorrelation	Calculate the cross-correlation of two one-dimensional sequences using the direct method of linear convolution.
ZeroPadder	Adjust the size of the input array of the input sequence to the next valid integer power of 2, and set the new tail element in the sequence to zero.

UnwrapPhase	Delete the discontinuities whose absolute values exceed PI or 180 degree and expand the phase array.
Scale	Delete the offset of the input signal x and scale the result so that the output sequence is in [- 1:1].
Normalize	Normalize the input vector according to the statistical distribution (μ, s) (μ is the mean and s is the standard deviation) to obtain the normalized vector with the statistical distribution of (0,1).
Upsample	Upsample the input sequence by the upsampling factor.
Downsample	Downsample the input sequence by downsampling factor and averaging.
Detrend	Detrend the input sequence.
WeightMovingAverage	Weighted average the input sequence is to smooth the fluctuation, which is similar to low-pass filtering.
ExponentialAverage	Exponentially average the input sequence is to smooth the fluctuation, which is similar to low-pass filtering.
Resampling	Resample the uniform input sequence with a new sampling rate, and the starting index position remains unchanged; Resample the non-uniform input sequence with a new sampling rate, and the starting index position remains unchanged.
SignalOperation.Digital	
GetDigitalSize	Returns the number of samples and signals contained in the digital input.
DigitalToBool	Converts a digital signal into a two-dimensional Boolean array.
BoolToDigital	Converts a two-dimensional Boolean array to a digital signal.
DigitalToBinary	Converts a digital signal to a one-dimensional unsigned byte integer array; Converts a digital signal to a one-dimensional ushort integer array; Converts a digital signal to a one-dimensional uint integer array.
BinaryToDigital	Converts a one-dimensional unsigned byte array to a digital signal; Converts a one-dimensional ushort array to a digital signal; Converts a one-dimensional uint array to a digital signal.
Shift	Shift the input digital signal.
Invert	Reverse the digital data in the input digital so that 0 convert 1 (or 1 convert 0).
GetSubset	Returns a subset of digital inputs used to plot extracted signals one by one or group on a graph.
AppendSignals	Combining two digital signals.
AppendSamples	Add all samples of the digital waveform src2 after the

	digital waveform src1.

2.4 SeeSharpTools.JY.GUI Class Library

The SeeSharpTools.JY.GUI class library currently contains 22 commonly used GUI controls in the field of test and measurement.

2.4.1 EasyChart

EasyChart is a custom control based on Microsoft MSChart control. It encapsulates the most commonly used single-channel or multi-channel continuous waveform, single-channel or multi-channel discrete waveform display functions. The included properties and methods are shown in the following table. See SeeSharpExamples/ Graphical User Interface/JYEasyChart Basics for examples. (The function of this control has been completely replaced by EasyChartX, and will not be updated later)

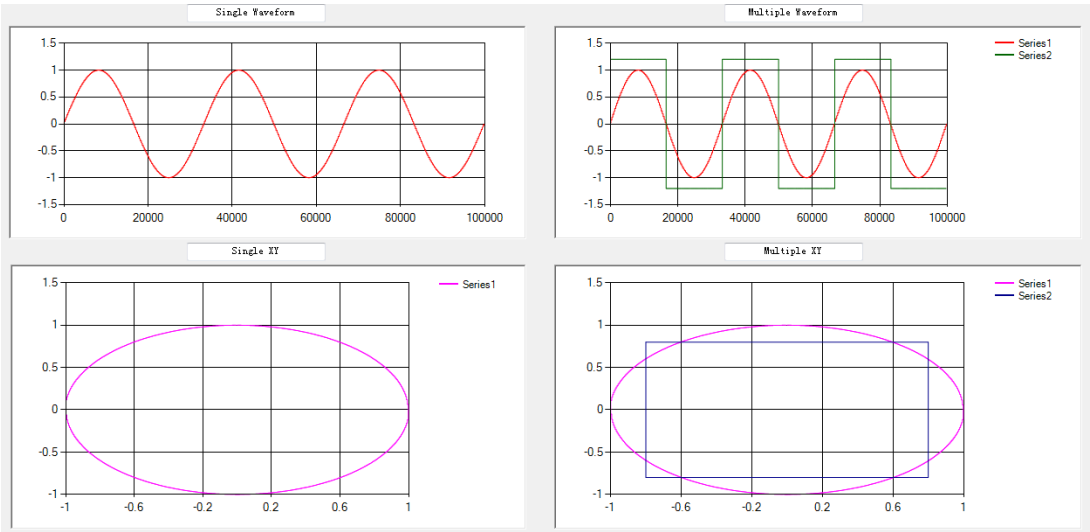
Properties	
Property Name	**Function Summary**
LegendVisible	Set whether to display the legend in Chart. The default is True.
Palette	Set a series of colors, which are used in turn for the color display of each channel waveform in the Chart.
XAxisLogarithmic[1]	Set whether the X axis is displayed in logarithm. The default is False.
YAxisLogarithmic[1]	Set whether the Y axis is displayed in logarithm. The default is False.
EasyChartBackColor	Set the color of the EasyChart frame
ChartAreaBackColor	Set the color of the EasyChart drawing area
LegendBackColor	Set the Legend background color
SeriesNames	Set the name of each line displayed in the Legend
YAutoEnable	Set whether the Y-axis range is automatically configured
AxisYMax	Set the maximum value of the Y axis
AxisYMin	Set the minimum value of the Y axis
Methods	

Plot	Displays single or multi-channel continuous waveforms, single or multi-channel discrete waveforms.
SaveAsCsv	Save the data in the current graph to the Csv file, with path entry and no arguments.
SaveAsImage	Save the current chart to the png file, with path entry and no parameter two overloads

Note 1: This property cannot be set at runtime. After this property is set to True, the corresponding data entered when calling the Plot method must be greater than zero.

After placing the EasyChart control on the Windows Form, you can select the EasyChart control and set the above properties in the property table and view the effects. Unless otherwise specified, these properties can also be set at runtime.

EasyChart supports single or multi-channel continuous or discrete waveform display. The continuous waveform here means that any two adjacent points on the waveform have the same interval on the X axis, which is usually used for time domain sampling waveform or spectrum display. When the Plot method is called to display continuous waveform, in addition to input waveform data. , can also set the starting position and interval on the X-axis; and discrete waveform refers to any point on the waveform, each is determined by a separate pair of x and y on the position of the Chart, in the call Plot method display For discrete waveforms, two arrays of equal length must be output, one-to-one corresponding to the position of each point on the Chart. For details, please refer to the EasyChart routine. The following figure shows the running effect of the EasyChart routine. The four graphs correspond to single-channel continuous waveform, multi-channel continuous waveform, single-channel discrete waveform and multi-channel discrete waveform.



2.4.2 AquaGauge

AquaGauge is a dashboard control for displaying values. Common properties are shown below:

Properties	
Property Name	**Function Description**
BackColor	Configure the dial background color.
Glossiness	Mirror effect debugging ratio (value of 0<=&&<=100)
Max	Dial maximum
Min	Dial minimum
TextDescription	Dial description
NumberOfDivisions	Number of dials separated, value range (1<=&&<=25)
NumberOfSubDivisions	Table plate separation number, value range (1<&&<=10 value)
Value	The current value displayed on the dial

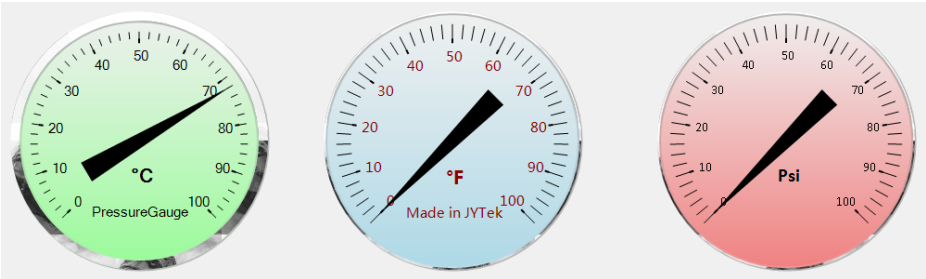
The control display effect is shown in the figure below. Please refer to the supporting example for the method used: SeeSharpExamples/ Graphical User Interface/AquaGuageControlExample.

2.4.3 PressureGauge

PressureGauge is a pressure dial control that functions like a dashboard. The display style is different. The common properties of the control are as follows:

Properties	
Property Name	**Function Description**
BackColor	Configure the dial background color
BorderWidth	Configure the thickness of the dial frame
Glossiness	Mirror effect debugging ratio (value of 0<=&&<=100)
Max	Dial maximum
Min	Dial minimum
DescriptionText	Dial description
UnitText	Dial unit description
NumberOfDivisions	Number of dials separated, value range (1<=&&<=25)
NumberOfSubDivisions	Table plate separation number, value range (1<&&<=10 value)
Value	The current value displayed on the dial

The display effect of the control is shown in the figure below. See the example for how to use it: SeeSharpExamples/ Graphical User Interface/PressureGaugeExample.



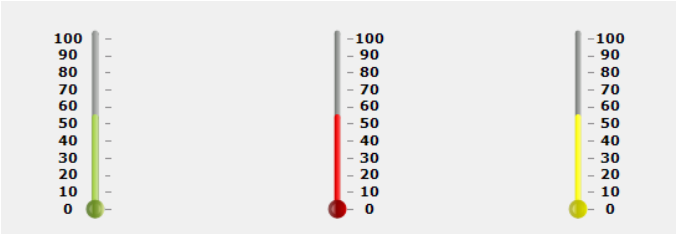
2.4.4 Thermometer

The Thermometer is a thermometer display control that can be used to display physical quantities such as temperature. The common properties of this control are shown in the following table.

Properties	
Property Name	**Function Description**
LineWidth	Configure the thermometer boundary line width
LineColor	Configure the color displayed by the thermometer
ForeColor	Configure the color of the thermometer boundary line
TickColor	Configure thermometer tick color
TickWidth	Configure thermometer tick line width

BallSize	Configure the size of the thermometer ball
TextStyle	Configure thermometer scale font style
TickStyle	Configure thermometer scale style
Value	Configure the display value of the thermometer
Min	Configure the minimum value displayed by the thermometer
Max	Configure the maximum value displayed by the thermometer
NumberOfDivision	Configure the number of thermometer scales
TextDecimals	Configure the number of decimal places displayed by the thermometer scale

The display effect of the control is as shown below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/ThermometerExample.

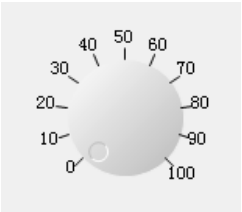


2.4.5 KnobControl

The KnobControl class is a knob control that performs numeric input operations. The common property methods for this control are shown in the following table.

Properties	
Property Name	**Function Description**
NumberOfDigits	The effective number of scientific expressions in numerical display (values of 0<=&&<=10)
Max	Knob maximum
Min	Knob minimum
NumberOfDivisions	The number of knobs is divided, the value range (values from 1 to 25)
TickVisible	Bool parameter, whether to display the knob value
Value	The current value represented by the knob
Events	
ValueChanged	The default event. Triggered when the user stops operating the knob.
ValueChanging	Triggered during user operation of the knob.

The display effect of the control is shown in the figure below. See the example for how to use it: SeeSharpExamples/ Graphical User Interface/KnobControlExamble.



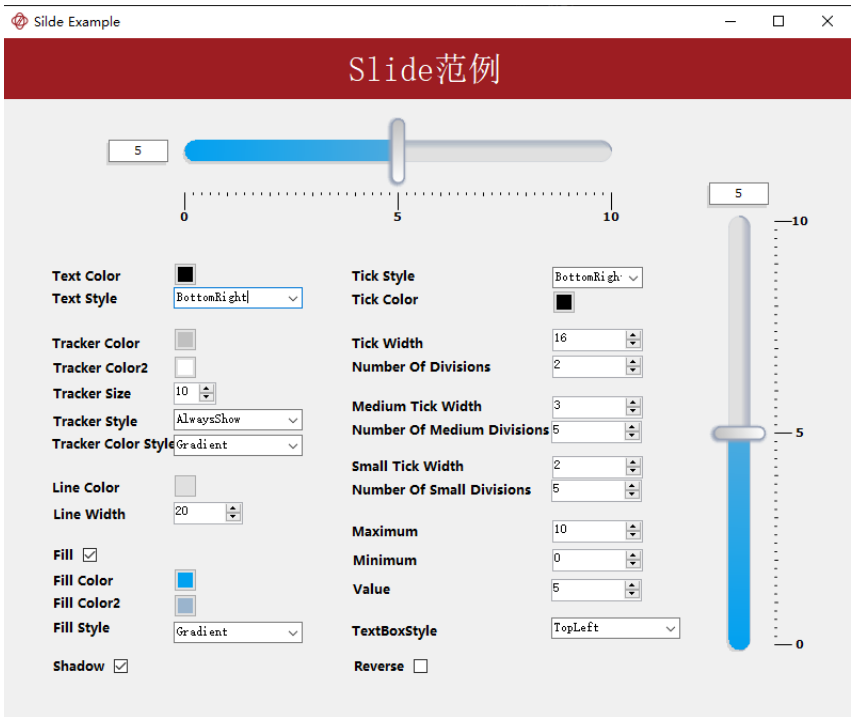
2.4.6 Slide

The Slide class is a commonly used slider control that performs numerical input operations. The common property methods for controls are shown in the following table.

Properties	
Properties	**Function Summary**
Value Decimals	The effective number of scientific expressions in numerical display (values of 0<=&&<=10)
Direction	Slider direction(Left to Right or Right to Left, Top to Bottom or Bottom to Top)
HeightIndent	Height indent between slider and control's selection box.
NumberOfDivisions	Number of divisions between the largest ticks, value range (1<=&&<=25). Use TickWidth to adjust the length of the largest ticks.
Orientation	Selection of the direction of the sliding rod (Vertical, Horizontal)
PaintShadow	Paint Shadows?
WidthIndent	Width indent between slider and control's selection box
NumberOfMediumDivisions	Number of divisions between the medium ticks. Use TickMediumWidth to adjust the length of the medium ticks
NumberOfSmallDivision	Number of divisions between the small ticks. Use TickSmallWidth to adjust the length of the small ticks
TickColor	Tick Color
TickIndent	Indent between ticks and slide bar
TickMediumWidth	Length of medium ticks
TickSmallWidth	Length of small ticks
TickStyle	Ticks' display style (None, TopLeft, BottomRight, Both)
TickWidth	Length of ticks
ForeColor	Slider Text color selection
TextDecimals	Decimal places for values texts
TextStyle	Value texts' style (None, TopLeft, BottomRight, Both)
TickTextIndent	Indent between ticks and value texts
Fill	Should the track line be color filled?
FillColor	Track line's fill color
FillSecondColor	Track line's fill color 2
FillStyle	Track line's fill style (Gradient, Aqua, Solid)

LineColor	Track line's main colors
LineWidth	Track line's height
TrackerColor	Slide block color
TrackerColorStyle	Slide block color style (None, Hide, AlwaysShow)
LargeChange	Large change for value. Press PgUp or PgDn to perform large value changes.
Max	Slider maximum
Min	Slider minimum
SmallChange	Small change for value. Press arrow keys to perform small value changes.
Value	The current value represented by the slider
Value Decimals	Effective decimal places used byvalue
TextBoxDecimal	Decimal places shown invalue text box
TextBoxFont	Font used in value text box
TextBoxIndent	Indent between slider and value text box
TextBoxStyle	Value text box style (None, TopLeft, BottomRight, Both)
Events	
ValueChanged	The default event. Triggered after the user operates the slider
ValueChanging	Triggered during user operation of the slider

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/SildeControlExample.



2.4.7 Led

The Led class is an LED light control that can perform the display operation of bool values. The commonly used property methods of the control are shown in the following table:

Properties	
Property Name	**Function Summary**
BlinkColor	Flashing color
BlinkInteral	The flashing time of the light, in milliseconds (100<=&&<=5000)
BlinkOn	Bool parameter, whether to turn on the blinking function
OffColor	Color selection when the light is off
OnColor	Color selection when the light is on
Style	LED Style Selection, Circular, Rectangular
Value	Bool parameters, lighting opening and closing options

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/SwitchControlExample.

2.4.8 EasyButton

The EasyButton class is a derivative of the Button. It can preset some images on the control to enhance the display of the Button. The related properties of the control are shown in the following table:

Properties	
Property Name	**Function Summary**
BackgroundStyle	Background style (Normal, Gradient). (3d only)
ButtonColor	Button's color. (3d only)
BorderColor	Border color. (3d only)
BorderRadius	Border radius. (3d only)
BorderThickness	Border thickness. (3d only)
Shadow	Paint Shadow? (3d only)
ThreeDimensional	Enables 3d

ButtonImage	Icon image
ImageSize	Size of ButtonImage
ImageStretch	Stretch image to the same size as font?
PreSetImage	Presets for ButtonImage

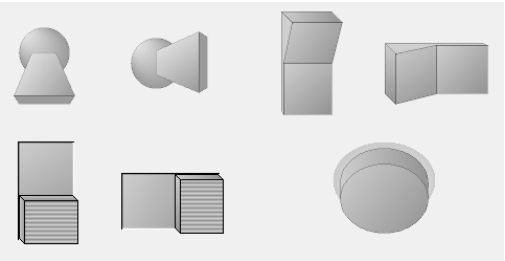
The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/SwitchControlExample

2.4.9 IndustrySwitch

The IndustrySwitch class is a control class for industrial switches. The common properties of the control are shown in the following table:

Properties	
Property Name	**Function Summary**
OffColor	Off state color
OnColor	On state color
Style	Multiple industrial Switch styles
OffText	Off state text
OnText	On state text
ShowText	Show text?
Value	Configure or get the value(On/Off) of the current control.
Events	
ValueChanged	Default event, triggered after the value is changed

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/SwitchControlExample.

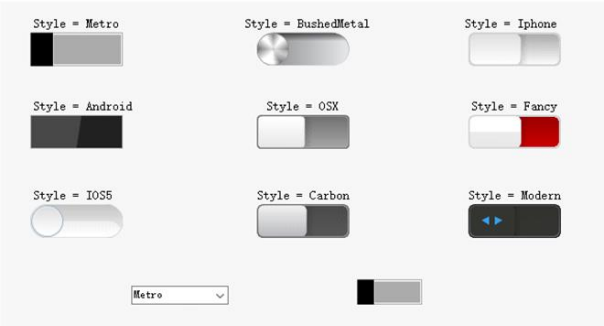


2.4.10 ButtonSwitch

The ButtonSwitch class is a control class for button switches, as shown in the following table:

Properties	
Property Name	**Function Summary**
Style	Multiple Switch check functions
Value	Bool parameters, Switch switch selection
Events	
ValueChanged	Default event, value change event

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/ButtonSwitchExample.



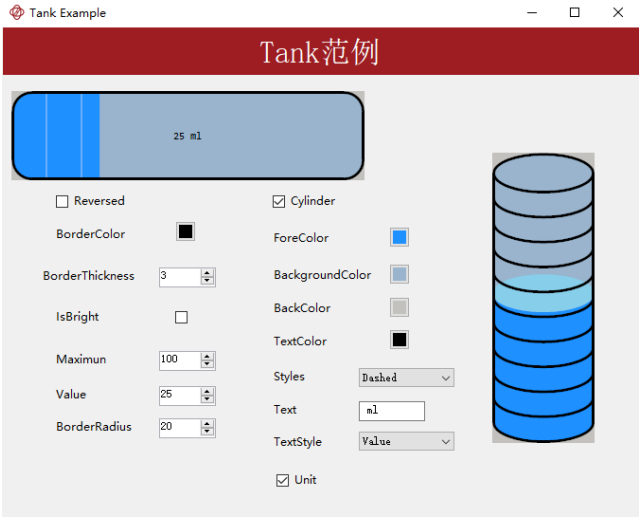
2.4.11 Tank

The Tank class is a jug-style control class. Common properties are shown in the following table:

Properties	
Property Name	**Function Summary**
BackgroundColor(Color)	Background color
BackgroundColorOpacity(Double)	Background color's opacity, ranging from 0 to 1.
BorderColor(Color)	Border color selection
BorderRadius(int)	Border radius
BorderWidth(int)	Border thickness
Cylindrical(bool)	Show tank as cylinder? Border radius needs to be > 0 to show as cylinder
DashedInterval(int)	Number of dashes
IsBright(bool)	Bool parameter, whether to use a bright color to display the Tank control
Orientation(Orientation)	Tank placement direction

Maximum(double)	Tank value maximum
Reversed(bool)	Whether the tank's water column direction reversed
Styles(TankStyles)	Tank style selection, Solid, Dashed
Text(String)	Text display (TextStyle needs to be Text)
TextColor(Color)	Tank text display color selection
TextStyle(TextStyleType)	Tank text display style selection (None, Percentage, Text, Value, ValueOverMaximum)
Value(double)	Current value represented by Tank
Unit(String)	Display unit. (TextStyle needs to be Value)

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/TankControlExample

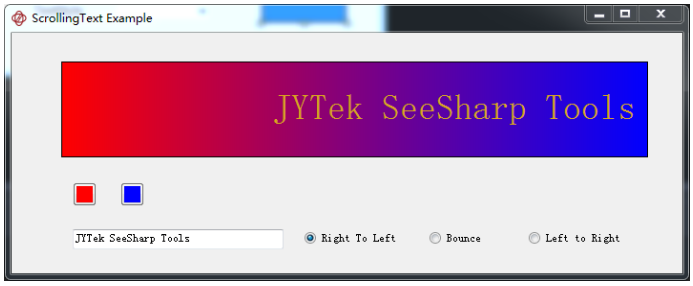


2.4.12 ScrollingText

The ScrollingText class is a control that scrolls through the display strings. Common properties are shown in the following table:

Properties	
Property Name	**Function Summary**
BorderColor	Border color selection
BorderThickness	Border thickness selection
ScrollDirection	Text scrolling direction selection
BorderVisible	Bool parameter, whether you need to display the border
Text	Scrolling description text
SrollSpeed	Scroll text speed (values of 1<=&&<=1000)
VerticleAligment	Set scroll bar text position

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/ScrollingTextExample.



2.4.13 SevenSegment

The SevenSegment class is a seven-segment digital tube control class. Common properties are shown in the following table:

Properties	
Property Name	**Function Summary**
BackColor	Background color
DarkColor	Digital tube dark color
LightColor	Digital tube lighting color
DecimalShow	Whether to display the decimal point position
ItalicFactor	Digital tube tilt coefficient
NumberOfChar	The total number of digital tubes
Value	Digital tube display content

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/SevenSegment.

2.4.14 SegmentBright

The SegmentBright class is a seven-segment digital tube control class. The display effect is different from that of SevenSegment. There are more color options. Common Properties are shown in the following table:

Properties

Property Name	Function Summary
BackColor1	Background color 1
BackColor2	Background color 2, mixed by background color 1 and background color 2 as a new background color
GradientBackground	Whether to display the gradient background color effect
BorderColor	Border color
BorderWith	Border width
DarkColor	When the digital tube is not displayed, the color
ForColor	Color selection when the digital tube is displayed
HighlightOpaque	Display translucent color, default is 50
NumberOfChar	Total number of digital tubes

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/SegmentBright.

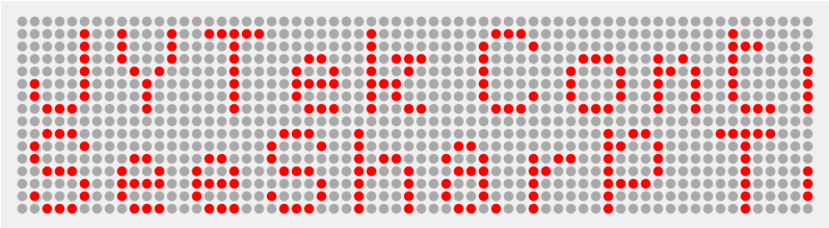
2.4.15 LedMatrixControl

The LedMatrixControl class is a Led matrix display control class. Common properties are shown in the following table:

Properties	
Property Name	**Function Summary**
LedOffColor	Color selection when LED matrix light is off
LedOnColor	Color selection when LED matrix light is on
LedStyle	LED matrix light style selection (circle, square)
LedRows	LED matrix light row number selection
LedColumns	LED matrix light column number selection
LedSizeCoeff	LED matrix light size factor (0<=&&<=1 value)
Items	Location configures the current display line Text position
	Direction configures the current display line Text scroll direction (Up, Down, Left, Right)
	Speed configures the current display line Text scrolling speed
Events	
StartMove	LED matrix light scrolling begins

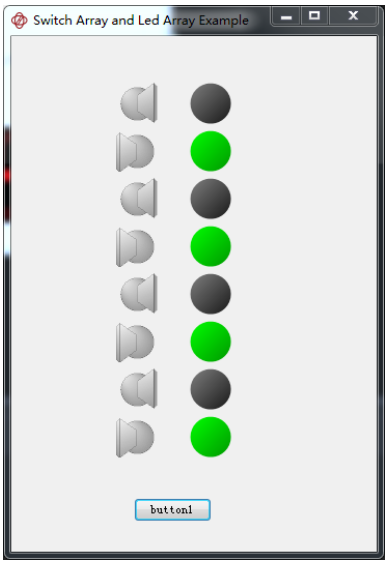
StopMove	LED matrix light scroll stop

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/LedMatrixControlExample.



2.4.16 SwitchArray and LEDArray

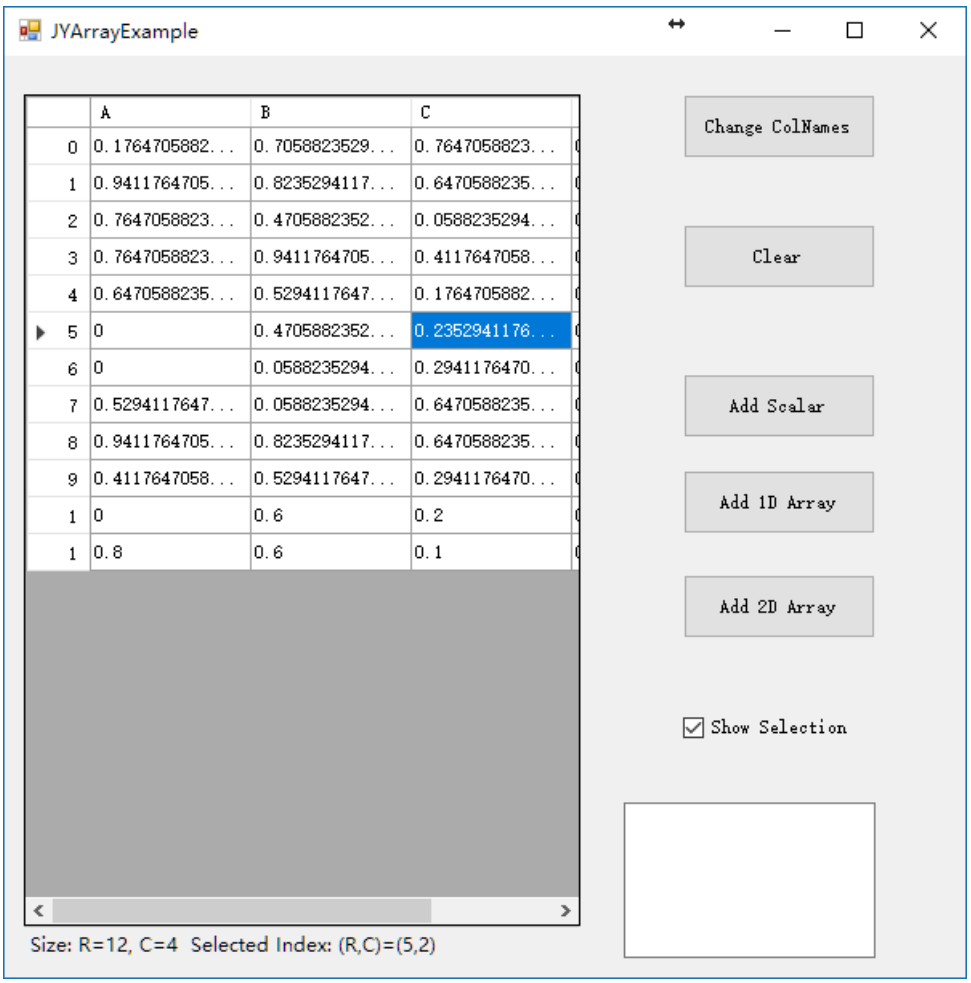
SwitchArray and LEDArray use the IndustrySwith control and LED control in SeeSharpTools.GUI as an array type. The control supports mouse control switch, program change value (single or array type), and the value change event enables the user interface to register. The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/JYGUIArrayExample.



2.4.17 JYArray

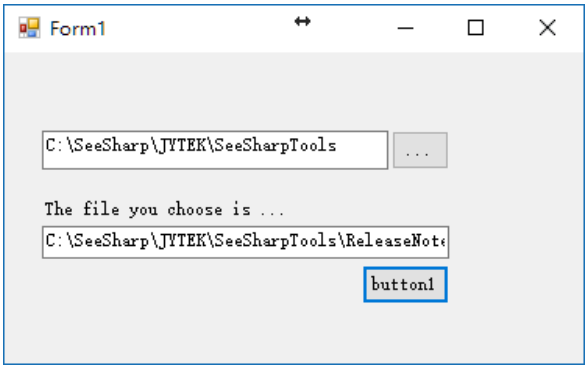
The JYArray class provides a two-dimensional array display interface to help measure data rendering. The control supports array accumulation display, change list name, display selection information, display field number, etc. At the same time, all attributes can be

extracted using attributes. The display effect of the control is as shown below.



2.4.18 PathControl

The PathControl class is a control class that receives the user configuration path. The PathControl provides four input modes, namely the browse button, text box input, mouse drag and drop, and program input. The path can be read out for use by the program. The display effect is shown below:



2.4.19 StripChart

StripChart is a custom control based on the Microsoft MSChart control. It implements the functions of single-channel and multi-channel small recorders. Common properties and methods are as follows:

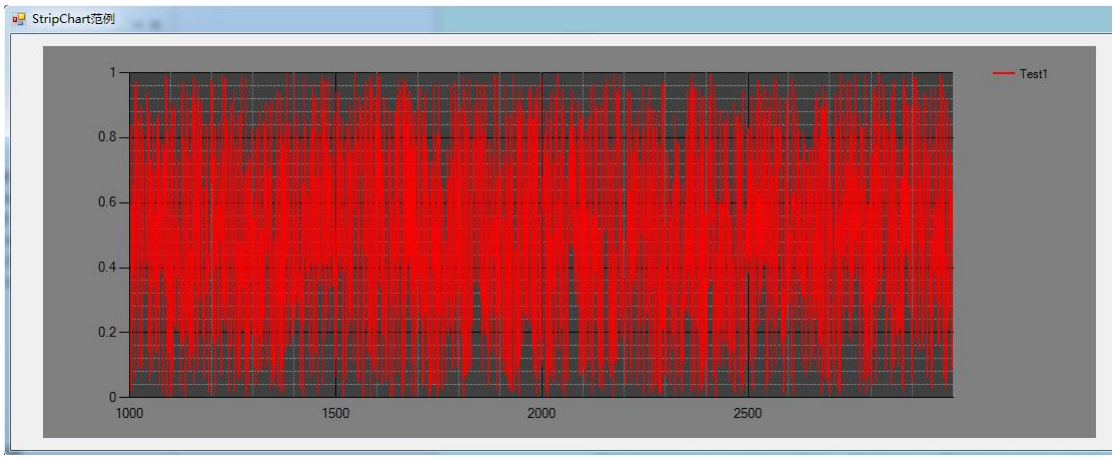
Properties	
Property Name	**Function Summary**
LegendVisible	Set whether to display the legend in Chart. The default is True.
Palette	Set a series of colors, which are used in turn to display the color of each channel waveform in Chart.
YAxisLogarithmic	Set whether the Y axis is displayed in logarithm. The default is False.
ChartBackColor	Set the StripChart frame color
ChartAreaBackColor	Set the color of the StripChart plot area
MajorGridEnabled	Set whether to enable the main grid line
MinorGridEnabled	Set whether to enable the auxiliary grid line
LineNum[1]	Set the number of lines to be displayed in the graph (Plot will automatically update to the actual number of lines)
LineWidth	Set the line width of each line
SeriesNames	Set the name of each line displayed in the Legend
YAutoEnable	Set whether the Y-axis range is automatically configured
AxisYMax	Set the maximum value of the Y axis
AxisYMin	Set the Y axis minimum
Displaydirection	Set the direction in which the recorder scrolls, from left to right or from right to left
DisPlayPoints[1]	Set the maximum number of points that can be displayed for each line in StripChart. After the number of points, the drawing area starts to scroll.
XAxisTypes	X-axis display type, which can be index, timestamp, user input value. The default is index.
TimeStampFormat	Set the display format of the timestamp
NextTimeStamp	Configure or get the next timestamp in timestamp mode
TimeInterval	Time difference between two samples in timestamp mode
XAxisStartIndex	Set the starting value of the index
XAxisTitle	X-axis title
XTitlePosition	X-axis title position
XTitleOrientation	X-axis title direction
YAxisTitle	Y-axis title
YTitlePosition	Y-axis title position

YTitleOrientation	Y-axis title direction
ScrollType	StripChart scroll mode
Methods	
Plot	Write multiple points to each line of StripChart.
PlotSingle	Write a point to each line of StripChart
Clear	Clear current drawing

Note 1: The attribute with 1 annotation cannot be set at runtime. After the property is set to True, the corresponding data input when calling the Plot method must be greater than zero.

After placing the StripChart control on the Windows Form, you can select the StripChart control and set the properties in the property table and view the effects. Unless otherwise specified, these properties can also be set at runtime.

StripChart supports a single write of multiple points (Plot) per line or a single point of each line (PlotSingle). The points after writing will be displayed in StripChart, when the number of points written per line exceeds the DisplayPoints configuration. The value of the plot area starts to scroll left or right, ensuring that the number of points per line in a single plot area does not exceed DisplayPoints. When the DisplayPoints is equal to 2000, total number of draw points is 3000, the display effect of StripChart is as follows. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/StripChartExamples.

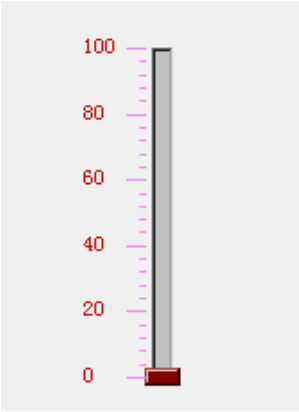


2.4.20 GaugeLinear

The GaugeLinear class is a slider control that implements numeric input operations. The control also supports setting the numerical slider/slider/scale color, and the scale of the control will automatically match according to the current space size. The property methods are shown in the following table.

Properties	
Property Name	**Function Summary**
BackColor	Configure the background color of the control
Maximum	Maximum value of the slider
Minimum	Minimum value of the slider
Orientation	Selection of sliding rod direction (Vertical, Horizontal)
SlideDirection	Slider value increment direction (LeftToRight, RightToLeft)
Value	Configure or get the value corresponding to the slider position
Event Name	**Event Description**
ValueChanged	Triggered when the value of the control is modified

The display effect of the control is shown in the figure below. For detailed usage, please refer to the supporting example: SeeSharpExamples/ Graphical User Interface/GaugeLinearExample.



2.4.21 EasyChartX

EasyChartX is a custom control based on Microsoft MSChart control. This control is an upgraded version of EasyChart. EasyChartX has added many common features of test measurement, simplified external interface, improved drawing efficiency, and added

partition view. Each line is displayed independently in a different drawing area. The properties, events, and methods of EasyChartX are shown in the following table:

Properties		
Property Name(Type)		**Function Summary**
LegendVisible(bool)		Set whether to display the legend in Chart. The default is True.
BackColor(Color)		Set the color of the EasyChartX frame
ChartAreaBackColor(Color)		Set the color of the EasyChartX plot area, which takes effect in both views.
AutoClear(bool)		Specify whether clear old datas before plot.
ForeColor(Color)		The font color of the axis Label and the name
Font(Font)		The font of the coordinate label
LegendBackColor(Color)		Background color of the legend
LegendFont(Font)		Legend font
LegendForeColor(Color)		The text color of the legend
SeriesCount(int)		Configure the number of lines displayed when not drawing, and get the number of lines in the current drawing.
GradientStyle(ChartGradientStyle)		Background color gradient type
SplitView(bool)		Whether the partition view is displayed, the default is false
Miscellaneous (MiscellaneousConfiguration)		Miscellaneous configuration
	MarkerSize(int)	Marker size, range of values [3-21]
	ShowFunctionMenu(bool)	Configure right click on the Chart, whether to display the function menu
	MaxSeriesCount(int)	The maximum number of lines, the default is 32, the data beyond this value is not displayed in the Chart
	MaxSeriesPointCount(int)	The maximum number of points displayed on a single line, the default is 4000. Increasing this value will increase the level of data display, but will reduce performance.
	CheckNaN(bool)	Whether to check the NaN data in the data. If not checked, Nan is displayed as an empty point; if checked, NaN is replaced with 1.5E-200 and displayed as an empty point.
	CheckNegtiveOrZero	Whether to check for non-positive numbers. If you do not check Nan, replace non-positive numbers with empty points; if you check Nan, replace non-positive numbers with 1E200 and 1E-200
	CheckInfinity	Whether to check the Inf data. If you do not check Nan, Inf is replaced with an empty point; if you check Nan, Inf is replaced with 1E200 and -1E200/1E-200.
	DataStorage(DataStorageType)	Configure the storage type of plot data, Including Clone And NoClone.
	Fitting(EasyChartX.FitType)	Screen selection algorithm selection. Use None to improve

		drawing speed when big data is available, but it may cause display errors in data trends.
	SplitLayoutDirection (LayoutDirection)	The layout direction of the partition view: from left to right or top to bottom
	SplitViewAutoLayout(bool)	Configure whether the partition view is automatically laid out. The default is true. When configured to false, DirectionChartCount will take effect.
	DirectionChartCount(int)	The number of drawing areas in the layout direction (SplitLayoutDirection is the number of drawing areas of a row when LeftToRight is used, and the number of drawing areas of a column when TopToDown is). This attribute takes effect when SplitViewAutoLayout is False.
	SplitLayoutColumnInterval(int)	The pixel interval between each column of the drawing area in the partition view (can be negative)
	SplitLayoutRowInterval(int)	The pixel interval between each drawing area of the partitioned view (can be negative)
LineSeries (design time name) Series (develop time name) (EasyChartXSeriesCollection)		Set the properties of each line. The number of configurations of this attribute can exceed the current number of lines. New lines are configured according to the corresponding index.
	Name(string)	Set the line name, which will be displayed in the legend
	Color(Color)	Line color
	Visible(bool)	Line visible or not
	Width(LineWidth)	Line width
	XPlotAxis(PlotAxis)	Which X axis is used, the current version can only use the main X axis.
	YPlotAxis(PlotAxis)	Which Y axis is used, the main Y axis and the sub Y axis can be selected
	Type(LineType)	Line type
	Marker(MarkerType)	Point marker type. Only valid when Type is Line
SplitPlotArea (EasyChartXPlotAreaCollection)		Properties of each drawing area when configuring and getting partitioned views
	BackColor(Color)	Drawing area background color
	AxisX(EasyChartXAxis)	Main X axis
	AxisX2(EasyChartXAxis)	Secondary X axis
	AxisY(EasyChartXAxis)	Main Y axis
	AxisY2(EasyChartXAxis)	Scondary Y axis
	XCursor(EasyChartXCursor)	X-axis cursor
	YCursor(EasyChartXCursor)	Y-axis cursor
Axes(EasyChartXAxis[]) AxisX(EasyChartXAxis) AxisX2(EasyChartXAxis) AxisY(EasyChartXAxis) AxisY2(EasyChartXAxis)		The coordinate axis configuration can be configured with related attributes of the X axis, the sub X axis, the Y axis, and the sub Y axis. This configuration is effective for both views.
	Name(string)	Get the axis name

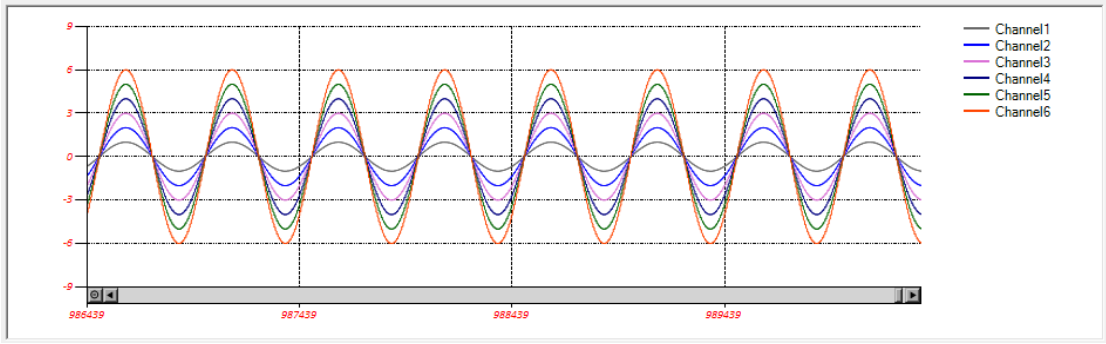
	Color(Color)	Configure axis color
	Title(string)	Configure axis title
	TitleOrientation (AxisTextOrientation)	Configure the axis title display direction
	TitlePosition (AxisTextPosition)	Configure the axis title display position
	LabelEnabled(bool)	Configure whether to display the value of the axis
	LabelFormat(string)	Configure the format of the label below the axis
	MajorGridEnabled(bool)	Main grid enable
	MajorGridColor(Color)	Main grid color
	MajorGridType(GridType)	Main grid line type
	MinorGridEnabled(bool)	Secondary mesh enable
	MinorGridColor(Color)	Secondary grid color
	MinorGridType(GridType)	Sub-grid line type
	TickWidth(double)	Configure the width of the marker line on the coordinate axis and the ratio of the height/width of the Chart
	AutoScale(bool)	Whether the axis range is automatically configured
	AutoScalingMode	When AutoScale is enabled, select ByGridCount to scale axis values according to grid count; select ByWholeNumbers to scale axis values by whole numbers
	Maximum(double)	Get the current axis maximum value. Configure the axis maximum when not drawing; when drawing and AutoScale is false, you can configure the axis maximum.
	Minimum(double)	Get the current coordinate axis minimum. Configure the minimum value of the coordinate axis when not drawing; when drawing and AutoScale is false, you can configure the minimum value of the coordinate axis.
	IsZoomed(bool)	Get the current axis is in the zoom state
	AutoZoomReset(bool)	Configure whether to cancel zooming each time you draw
	InitWithScaleView(bool)	Configure whether to use the zoom view by default when drawing starts
	ViewMaximum(double)	Get the maximum value of the current axis zoom view. When InitWithScaleView is true, configure the axis zoom view maximum when starting drawing
	ViewMinimum(double)	Get the current coordinate axis zoom view minimum value, when InitWithScaleView is true, configure the coordinate axis to zoom the view minimum when starting drawing
	IsLogarithmic(bool)	Whether the current axis uses the logarithmic display.
	LogarithmBase(double)	Under logarithmic display, the used logarithmic base.
	LogLabelStyle	Under logarithmic display, the label style (F0, E2).
	ShowLogarithmicLines(bool)	Under logarithmic display, show visual helper lines?
	Zoom(double max, double min)	Scale the current axis to a range of two values
	ZoomReset(int resetTimes)	Rollback axis scaling operation
Cursors(EasyChartXCursor[]) XCursor(EasyChartXCursor) YCursor(EasyChartXCursor)		Cursor configuration, you can configure the X cursor, Y cursor related properties. This configuration takes effect in both views.

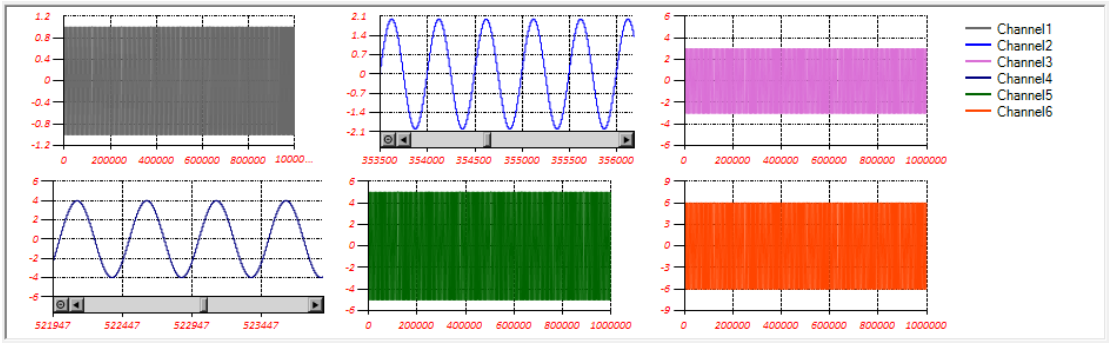
	Name(string)	Get cursor name
	Mode(CursorMode)	Cursor types, including non-enable, cursor, and region selection
	Color(Color)	Cursor color
	SelectionColor(Color)	The color of the area when the cursor area is selected
	AutoInterval(bool)	Whether to automatically configure the selection interval
	Interval(double)	Cursor selection interval when AutoInterval
	Value(double)	Get the current position of the cursor when in cursor mode
TabCursors(TabCursorCollection)		Collection of label cursors
	CursorValueFormat(string)	Displays the format of the value when the mouse is over the label cursor
	void Add(TabCursor item)	Add tag cursor
	void Clear()	Clear label cursor
	TabCursor(Element in TabCursors)	Label cursor class
	TabCursor.Color(Color)	Cursor color
	SeriesIndex(int)	The index number of the curve bound to TabCursor, less than 0 is unbound.
	TabCursor.Value(double)	The value of the X axis to which the cursor is currently aligned
	TabCursor.YValue(double)	Whe the Direction is Vertical, this property is readonly and maintain the Y value of the point that the X value and SeriesIndex bound to. Available only when SeriesIndex is valid. When the Direction is Horizental, this value can be read and write.
	TabCursor.Direction (TabCursorDirection)	The direction of tab cursor. The value can be Horizental or Vertical. The tab cursor cannot trace chart point and XValue/SeriesIndex property will be invalid when the value is Horizental.
	TabCursor.Name(string)	Cursor name
	TabCursor.Enabled(bool)	Display cursor or not
Custom Events		
AxisViewChanged		This event is triggered when the range of the axis changes
	EasyChartXViewEventArgs	Event parameters: Axis: the axis that triggered the event ParentChart: The EasyChartX instance that triggered the event IsScaleViewChanged: Whether it is an event triggered by a zoom view change IsRaisedByMouseEvent: Whether it is an event triggered by a mouse operation
CursorPositionChanged		The event is triggered when the user uses the cursor to align to a line.
	EasyChartXCursorEventArgs	Event parameters: Cursor: Cursor that triggers an event ParentChart: The EasyChartX instance that triggered the event SeriesIndex: the line index that triggers the Cursor event

		IsRaisedByMouseEvent: Whether the mouse action is triggered
TabCursorChanged		This event is triggered after the TabCursor is operated or moved.
	TabCursorEventArgs	Event parameters: Cursor: TabCursor that triggered the event ParentChart: The EasyChartX that triggered the event Operation: Event trigger action type: value is modified, Cursor is added, Cursor is deleted ParentChartAre: the drawing area object where TabCursor is located
BeforePlot		Trigger the event before drawing the curve in the drawing area
	EasyChartXPlotEventArgs	Event parameters: IsClear: Whether the event is triggered by Clear ParentChart: The EasyChartX handle that triggered the event
AfterPlot		Trigger the event after drawing the curve in the drawing area
	EasyChartXPlotEventArgs	IsClear: Whether the event is triggered by Clear ParentChart: The EasyChartX handle that triggered the event
Methods		
Plot<TDataType>(…)		Display single-channel or multi-channel continuous waveform, single-channel or multi-channel discrete waveform. The overload of the two-dimensional array of this method contains the parameters of majorOrder(MajorOrder) to configure whether to draw by row or column. Generic overload support for the Plot method: double/float/int/uint/short/ ushort/byte data.
AddDataMarker(…)		Add data markers to the current drawing. These markers will be cleared the next time Plot or Clear. A single drawing can add Marker multiple times.
SaveAsCsv(string filePath)		Save the data in the current graph to the Csv file, with path entry and no arguments.
SaveAsImage(string filePath)		Save the current chart to the png file, with path entry and no parameter two overloads
ZoomReset(double min, double max)		Stop scaling all axes
IsPlotting()		Get the state of chart. This method will return false when chart is not plotting (Plot method has not been called or Clear operation has just been executed.)
Clear()		Clear all the plotting data in the chart.

After placing the EasyChartX control on the Windows Form, you can select the EasyChartX control and set the above properties in the property sheet. EasyChartX supports single or multi-channel continuous or discrete waveform display. The value range

of the waveform should be [-2.5E28 , 2.5E28], beyond this range will cause the Chart to throw a runtime exception. The continuous waveform here means that any two adjacent points on the waveform have the same interval on the X axis, usually used for time domain sampling waveforms. Or the spectrum display, when the Plot method is called to display the continuous waveform, in addition to the input waveform data, the starting position and interval on the X-axis can be set; and the discrete waveform refers to any point on the waveform, both of which are independent pairs. x and y to determine its position on the Chart, when calling the Plot method to display discrete waveforms, you must output two arrays of equal length, one-to-one corresponding to the position of each point on the Chart. Refer to the EasyChartX routine for detailed usage. The figure below is a set of sinusoidal signals in the EasyChartX mode. For detailed usage, please refer to the supporting example: SeeSharpExamples/ EasyChartX Example.



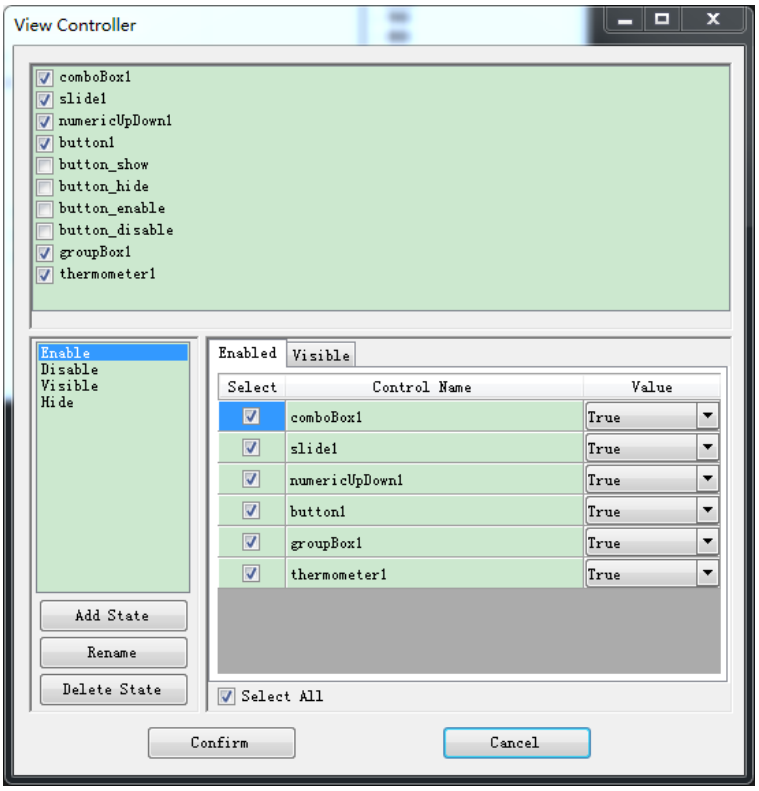


2.4.22 ViewController

ViewController is a class used to manipulate the interface linkage. The state of the interface control is adjusted according to the current state by switching the state. Currently, the Enabled (enabled) and Visible properties (visible) of the operation control are supported. The properties of the control are as follows: It shows that other properties except State are configured at design time. The control is only visible at design time and is automatically hidden at runtime.

Properties	
Property Name	**Function Summary**
StateNames(string[])	All state names
State(string)	Current state, case insensitive
ControlInfos(string[])	Control information for each state without manual modification

ViewController's StateNames and ControlInfos are configured through the small triangle of the ViewController control. The configuration interface after clicking is shown in the figure below. For detailed usage, see the example: SeeSharpExamples/ Graphical User Interface/ViewControllerExample.



The process of configuring status and linkage information is as follows:

- First select the controls that need to be linked in the top list. These controls will be automatically synchronized to the table below.

- Add, rename, or delete the status at the bottom left. To configure the control properties of a state, click on the status.
- On the right side, tick the control that needs to be operated, and then select the corresponding Enabled and Visible values. Enabled and Visible switch through the tab page above.
- Click Confirm when the configuration is complete.

When used, you can use the method similar to viewController1.State = "idle" to gray out or hide the controls in the idle configuration that are not enabled or not displayed. This property is cross-thread safe and does not require an additional Invoke to be called when the non-UI thread executes.

2.4.23 StripChartX

StripChartX is a recorder control based on Microsoft MSChart control. This control is an upgraded version of StripChart. StripChartX has added many common features of test measurement, simplified external interface, improved drawing efficiency, and added partition view. Each line is displayed independently in a different drawing area. The properties, events, and methods of StripChartX are shown in the following table:

Properties	
Property Name(类型)	**Function Summary**
LegendVisible(bool)	Set whether to display the legend in Chart. Default is True.
BackColor(Color)	Set the StripChartX frame color
ChartAreaBackColor(Color)	Set the color of the StripChartX plot area, which takes effect in both views.
ForeColor(Color)	The font color of the axis Label and the name
Font(Font)	The font of the coordinate label
LegendBackColor(Color)	Background color of the legend
LegendFont(Font)	Legend font
LegendForeColor(Color)	Legend font color
SeriesCount(int)	Configure the number of lines displayed when not drawing, and get the number of lines in the current drawing.
GradientStyle(ChartGradientStyle)	Background color gradient type
ScrollType(StripScrollType)	Scroll type. Cumulative type (integrated to the maximum number of points and then scrolled) / scroll type (initial or scrolling)
DisplayPoints(int)	The recorder displays the maximum number of points, and the data exceeding the number is discarded.

XDataType(XAxisDataType)		X-axis data type: index / timestamp / string
TimeStampFormat		Timestamp format, valid when XDataType is timestamp
NextTimeStamp		The first timestamp of the next drawing, valid when XDataType is timestamp
TimeInterval(TimsSpan)		Time interval, valid when XDataType is timestamp
StartIndex(int)		Start index number, valid when XDataType is index
Direction(ScrollDirection)		Scroll direction, right to left / left to right
Miscellaneous (MiscellaneousConfiguration)		Miscellaneous configuration
	MarkerSize(int)	Marker size, range of values [3-21]
	ShowFunctionMenu(bool)	Configure right click on the Chart, whether to display the function menu
	MaxSeriesCount(int)	The maximum number of lines, the default is 32, the data beyond this value is not displayed in the Chart
	MaxSeriesPointCount(int)	The maximum number of points displayed on a single line, the default is 4000. Increasing this value will increase the level of data display, but will reduce performance.
	CheckNaN(bool)	Whether to check the NaN data in the data. If not checked, Nan is displayed as an empty point; if checked, NaN is replaced with 1.5E-200 and displayed as an empty point.
	CheckNegtiveOrZero	Whether to check for non-positive numbers. If you do not check Nan, replace non-positive numbers with empty points; if you check Nan, replace non-positive numbers with 1E200 and 1E-200
	CheckInfinity	Whether to check the Inf data. If you do not check Nan, Inf is replaced with an empty point; if you check Nan, Inf is replaced with 1E200 and -1E200/1E-200.
	Fitting(StripChartX.FitType)	Screen selection algorithm selection. Use None to improve drawing speed when big data is available, but it may cause display errors in data trends.
	SplitLayoutDirection (LayoutDirection)	The layout direction of the partition view: from left to right or top to bottom
	SplitViewAutoLayout(bool)	Configure whether the partition view is automatically laid out. The default is true. When configured to false, DirectionChartCount will take effect.
	DirectionChartCount(int)	The number of drawing areas in the layout direction (SplitLayoutDirection is the number of drawing areas of a row when LeftToRight is used, and the number of drawing areas of a column when TopToDown is). This attribute takes effect when SplitViewAutoLayout is False.
	SplitLayoutColumnInterval(int)	The pixel interval between each column of the drawing area in the partition view can be negative
	SplitLayoutRowInterval(int)	The pixel interval between each drawing area of the partitioned view can be negative
LineSeries (design time attribute name) Series (development time attribute		Set the properties of each line. The number of configurations of this attribute can exceed the current number of lines. New lines are configured according to the corresponding index.

name) (EasyChartXSeriesCollection)		
	Name(string)	Set the line name, which will be displayed in the legend
	Color(Color)	Line color
	Visible(bool)	Line visible or not
	Width(LineWidth)	Line width
	XPlotAxis(PlotAxis)	Which X axis is used, the current version can only use the main X axis.
	YPlotAxis(PlotAxis)	Which Y axis is used, the main Y axis and the sub Y axis can be selected
	Type(LineType)	Line type
	Marker(MarkerType)	Point marker type. Only valid when Type is Line
SplitPlotArea (EasyChartXPlotAreaCollection)		Properties of each drawing area when configuring and getting partitioned views
	BackColor(Color)	Drawing area background color
	AxisX(EasyChartXAxis)	Main X axis
	AxisX2(EasyChartXAxis)	Secondary X axis
	AxisY(EasyChartXAxis)	Main Y coordinate axis
	AxisY2(EasyChartXAxis)	Secondary Y axis
	XCursor(EasyChartXCursor)	X-axis cursor
	YCursor(EasyChartXCursor)	Y-axis cursor
Axes(EasyChartXAxis[]) AxisX(EasyChartXAxis) AxisX2(EasyChartXAxis) AxisY(EasyChartXAxis) AxisY2(EasyChartXAxis)		The coordinate axis configuration can be configured with related attributes of the X axis, the sub X axis, the Y axis, and the sub Y axis. This configuration is effective for both views.
	Name(string)	Get the axis name
	Color(Color)	Configure axis color
	Title(string)	Configure axis title
	TitleOrientation (AxisTextOrientation)	Configure the axis title display direction
	TitlePosition (AxisTextPosition)	Configure the axis title display position
	LabelEnabled(bool)	Configure whether to display the value of the axis
	LabelFormat(string)	Configure the format of the label below the axis
	MajorGridEnabled(bool)	Main grid enable
	MajorGridColor(Color)	Main grid color
	MajorGridType(GridType)	Main grid line type
	MinorGridEnabled(bool)	Secondary mesh enable
	MinorGridColor(Color)	Secondary grid color
	MinorGridType(GridType)	Sub-grid line type
	TickWidth(double)	Configure the width of the marker line on the coordinate axis and the ratio of the height/width of the Chart

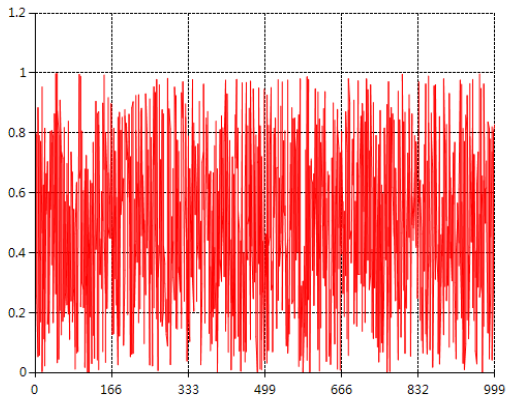
	AutoScale(bool)	Whether the axis range is automatically configured
	Maximum(double)	Get the current axis maximum value. Configure the axis maximum when not drawing; when drawing and AutoScale is false, you can configure the axis maximum.
	Minimum(double)	Get the current coordinate axis minimum. Configure the minimum value of the coordinate axis when not drawing; when drawing and AutoScale is false, you can configure the minimum value of the coordinate axis.
	IsZoomed(bool)	Get the current axis is in the zoom state
	AutoZoomReset(bool)	Configure whether to cancel zooming each time you draw.
	InitWithScaleView(bool)	Configure whether to use the zoom view by default when drawing starts.
	ViewMaximum(double)	Get the maximum value of the current axis zoom view. When InitWithScaleView is true, configure the axis zoom view maximum when starting drawing.
	ViewMinimum(double)	Get the current coordinate axis zoom view minimum value, when InitWithScaleView is true, configure the coordinate axis to zoom the view minimum when starting drawing
	IsLogarithmic(bool)	Whether the current axis uses the exponential display. Only the Y axis is supported.
	Zoom(double max, double min)	Scale the current axis to a range of two values
	ZoomReset(int resetTimes)	Rollback axis scaling operation
Cursors(EasyChartXCursor[]) XCursor(EasyChartXCursor) YCursor(EasyChartXCursor)		Cursor configuration, you can configure the X cursor, Y cursor related properties. This configuration takes effect in both views.
	Name(string)	Get the cursor name
	Mode(CursorMode)	Cursor types, including non-enable, cursor, and region selection
	Color(Color)	Cursor color
	SelectionColor(Color)	Select the color of the area when the cursor area is selected
	AutoInterval(bool)	Whether to automatically configure the selection interval
	Interval(double)	Cursor selection interval when AutoInterval
	Value(double)	Get the current position of the cursor when in cursor mode
TabCursors(TabCursorCollection)		Collection of label cursors
	CursorValueFormat(string)	Displays the format of the value when the mouse is over the label cursor
	void Add(TabCursor item)	Add tag cursor
	void Clear()	Clear tag cursor
	TabCursor(Element in TabCursors)	Label cursor class
	TabCursor.Color(Color)	Cursor color
	SeriesIndex(int)	The index number of the curve bound to TabCursor, less than 0 is unbound.
	TabCursor.Value(double)	The value of the X axis to which the cursor is currently aligned
	TabCursor.YValue(double)	The X axis of the cursor alignment corresponds to the Y

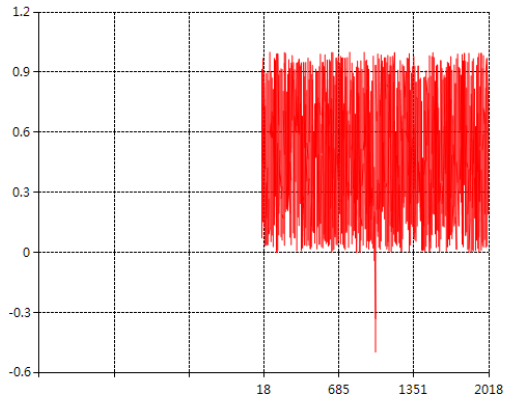
		value of the curve. Available only when SeriesIndex is valid.
	TabCursor.Name(string)	Cursor name
	TabCursor.Enabled(bool)	Display cursor or not
Custom Event		
AxisViewChanged		This event is triggered when the range of the axis changes
	EasyChartXViewEventArgs	Event parameters: Axis: the axis that triggered the event ParentChart: The EasyChartX instance that triggered the event IsScaleViewChanged: Whether it is an event triggered by a zoom view change IsRaisedByMouseEvent: Whether it is an event triggered by a mouse operation
CursorPositionChanged		The event is triggered when the user uses the cursor to align to a line.
	EasyChartXCursorEventArgs	Event parameters: Cursor: Cursor that triggers an event ParentChart: The EasyChartX instance that triggered the event SeriesIndex: the line index that triggers the Cursor event IsRaisedByMouseEvent: Whether the mouse action is triggered
BeforePlot		Trigger the event before drawing the curve in the drawing area
	EasyChartXPlotEventArgs	Event parameters: IsClear: Whether the event is triggered by Clear ParentChart: The EasyChartX handle that triggered the event
AfterPlot		Trigger the event after drawing the curve in the drawing area
	EasyChartXPlotEventArgs	IsClear: Whether the event is triggered by Clear ParentChart: The EasyChartX handle that triggered the event
Method		
Plot<TDataType>(…)		Record multiple samples of single or multiple channels. When the number of samples in the recorder exceeds the value of the DisplayPoints attribute, the previous points are discarded. Generic overload support for the Plot method: double/float/int/uint/short/ushort/byte data.
PlotSingle<TDataType>(…)		Record a single or multi-channel sample. When the number of samples in the recorder exceeds the DisplayPoints property value, the previous point is discarded. Generic overload support for the PlotSingle method: double/float/int/uint/short/ushort/byte data.
SaveAsCsv(string filePath)		Save the data in the current graph to the Csv file, with path entry and no arguments.

SaveAsImage(string filePath)	Save the current chart to the png file, with path entry and no parameter two overloads
ZoomReset(double min, double max)	Stop scaling all axes
Clear()	Clear all current drawings

After placing the StripChartX control on the Windows Form, you can select the StripChartX control and set the above properties in the property table. StripChartX supports single or multi-channel continuous or discrete waveform record display, the value range of the waveform should be in [-2.5 E28, 2.5E28], beyond this range will cause the Chart to throw a runtime exception.

StripChartX has two scroll modes: accumulation mode and scroll mode. The number of points in the accumulation mode accumulates until the value exceeds the value of DisplayPoints. The scroll mode starts scrolling from the beginning side, and the number of points is less than the configuration of DisplayPoints. The display is blank. The display effects of the two modes are as follows:





2.5 SeeSharpTools.JY.Database Class Library

JY.Database contains a DbOperation class, which encapsulates common operations on common databases, supported database types such as Microsoft SQL Server, Microsoft Access, MySql, Oracle, OleDb, ODBC, etc. Users do not need to pay attention to the actual differences of the database, only need to provide Correspond to the database connection string and database type, and then use the general SQL statement

to access the database. The methods in the class are shown in the following table:

Method Name	Function Summary
DbOperation Class	
ExecuteNonQuery	Generally used for UPDATE, INSERT, and DELETE statements. The only return value is the number of rows affected by the command. If the number of rows affected is 0, the operation is unsuccessful. If it is greater than 0, the operation is successful.
ExecuteScalar	Execute the query and return a result or a column returned by the query, such as the number of records in the given table or the current server time. The return value is an object, which can be cast to the appropriate type as needed.
ExecuteDataReader	The DataReader is the easiest and most efficient way to select some data from the data source. The returned DatReader object provides a "cursor" form of reading. When the row is read from the result, the "cursor" will Continue reading to the next line, returning a row of data for each call.
ExecuteDataTable	Execute a query command, the returned DataTable object is very similar to the actual physical database table, it consists of a set of columns of specific attributes, may contain 0 rows or rows of data. The data table can also define the primary key (may be One or more columns), columns can also contain constraints.

2.6 SeeSharpTools.JY.Localization Class Library

JY.Localization contains a Localization class, which is mainly used to localize WinForms programs. It needs to be combined with resource files in the project. The methods included in the class are as follows:

Method Name	Function Summary
Localization Class	
SetLang	Read the information in the resource file of the corresponding language, and implement the localization operation on the interface of the specified form.

2.7 SeeSharpTools.JY.DSP.FilterMCR Class Library

JY.DSP.Fundamental contains three static classes: IIRFilter, FIRFilter and JYSpectrum, which provide IIR filter, FIR filter, and spectrum calculation functions respectively. The method is shown in the following table:

Method Name	Function Summary
IIRFilter Class/FIR Class	

Initialize	Preloading the function that the class needs to call. After the loading is completed, the first call time will be greatly shortened.
ProcessLowpass	Low pass filter.
ProcessHighpass	High pass filter.
ProcessBandpass	Bandpass filter.
ProcessBandstop	Band stop filter.
JYSpectrum Class	
FFTSpectrum	Calculate the spectrum of the input time domain signal and set the unit of the output spectrum;

Precautions:

1. Project Properties -> Debug -> Enable Debugger, please uncheck "Enable Visual Studio Hosting Process".
2. MCR(R2017a (9.2) 64-bit) and the update package should be installed before running the application that has referenced this library. The installers can be acquired in the following link: MCR installer， MCR update package.
3. DSPMatlab.dll and MWArray.dll should be copied to execution directory before execution. This operation is always done when compiling the project. While sometimes these two files are not copied to execution directory and this will lead to a FileNotFoundException. Copy DSPMatlab.dll and MWArray.dll from SeeSharpTools installer directory to the execution path will solve this problem.

2.8 SeeSharpTools.JY.DSP.SoundVibration Class Library

JY.DSP.SoundVibration contains a HarmonicAnalyzer static class, mainly for the analysis and processing of sound and vibration signals. The methods included in the class are as follows:

Method Name	**Function Summary**
HarmonicAnalyzer Class	
ToneAnalysis	Calculate the fundamental and all harmonic frequencies and amplitudes of the input signal, as well as total harmonic distortion (THD). This function has two overloads: the first overload can calculate the overall harmonic analysis result and the second overload can calculate the power level of each harmonic wave.

The ToneAnalysisResult class is a class that saves the calculation results of

waveform indicators. The properties are as follows.

Property Name	Function Summary
ToneAnalysisResult Class	
THD	Harmonic distortion
THDPlusN	Total harmonic distortion plus noise
SINAD	AC signal to noise ratio
SNR	Signal to noise ratio
NoiseFloor	Noise floor
ENOB	Effective number of bits

2.9 SeeSharpTools.JY.DSP.Utility Class Library

JY.DSP.Utility contains 7 static classes: Phase, Synchronizer, ToneAnalysis, SystemNoiseCalculation, PeakValleyAnalysis, SignalProcess, MedianFilter and ToneAnalyzer. Among them, PeakSprectrum is used to detect the peak power and frequency of the signal; Phase is used to detect the phase information of the signal; Synchronizer Class is used to compensate the phase of multi-channel data; ToneAnalysis calculates the harmonic/signal-to-noise ratio of the waveform; PeakValleyAnalysis is used to detect the peaks and troughs of the waveform; SignalProcess is used to detect the peak value of the waveform, zero-crossing detection The methods included in the upper and lower limit detection; MedianFilter is a non-linear digital filtering technique, often used to remove noise from a signal; ToneAnalyzer is designed for single-tone wave form analyzis.

The interface of these class are as follows:

Method Name	Function Summary
Phase Class	
CalPhaseShift	Calculate the phase difference between waveforms
Synchronizer Class	
Sync	Compensate the phase of multi-channel data
SystemNoiseCalculation Class	
CaculateSystemNoise	Calculate the RMS of the noise over the specified bandwidth
PeakValleyAnalysis Class	
FindPeaks	Detecting the peak of the signal
FindValleys	Detecting the valley of the signal

The SignalProcess class provides threshold detection function and AC/DC rms calculation function. The functions are listed below:

SignalProcess Class

CheckThreshold	Search for the peak/valley value of the threshold that exceeds the threshold.
CheckCrossZeroPoints	Search for values in the waveform that climb above/fall below zero. Return a dictionary of indexes and values
CheckInRange	Detects the value of the waveform that exceeds the range. The return value is a List whose length is equal to the length of the waveform. The result is saved according to the index: 1 is the upper limit, -1 is lower than the lower limit, and 0 is within the range.
EstimateACDC	Calculate the AC and DC components of the signal

SignalOperation provides signal filtering, operating on analog signals, operating on digital signals, and signal conversion.The functions of this class are listed below.

SignalOperation 类		
Filtering		Signal Filtering Class
	FIRFilter	Filter the input sequence using a direct FIR filter specified by the fir coefficient.
	IIRFilter	Filter the input sequence x by a direct type IIR filter specified by the reverse coefficient and the forward coefficient.
	MedianFilter	Median filter the input sequence.
	ZeroPhaseFilter	Filter the input sequence using zero phase filter without boundary extension.
Analog		Analog Operation Class
	AutoCorrelation	Calculate the autocorrelation of one-dimensional sequence.
	Convolution	Calculate the convolution of two one-dimensional input sequences using direct method of linear convolution.
	Invert	Calculate the opposite number for each element in the array.
	CrossCorrelation	Calculate the cross-correlation of two one-dimensional sequences using the direct method of linear convolution.
	ZeroPadder	Adjust the size of the input array of the input sequence to the next valid integer power of 2, and set the new tail element in the sequence to zero.

	UnwrapPhase	Delete the discontinuities whose absolute values exceed PI or 180 degree and expand the phase array.
	Scale	Delete the offset of the input signal x and scale the result so that the output sequence is in [–1:1].
	Normalize	Normalize the input vector according to the statistical distribution (μ, s) (μ is the mean and s is the standard deviation) to obtain the normalized vector with the statistical distribution of (0,1).
	UpSample	Upsample the input sequence by the upsampling factor.
	Downsample	Downsample the input sequence by downsampling factor and averaging.
	Detrend	Detrend the input sequence.
	WeightMovingAverage	Weighted average the input sequence is to smooth the fluctuation, which is similar to low-pass filtering.
	ExponentialAverage	Exponentially average the input sequence is to smooth the fluctuation, which is similar to low-pass filtering.
	Resampling	Resample the uniform input sequence with a new sampling rate, and the starting index position remains unchanged.
Digital		Digital Operation Class
	GetDigitalSize	Returns the number of samples and signals contained in the digital input.
	DigitalToBool	Converts a digital signal into a two-dimensional Boolean array.
	BoolToDigital	Converts a two-dimensional Boolean array to a digital signal.
	DigitalToBinary	Converts a digital signal to a one-dimensional unsigned byte integer array.
	BinaryToDigital	Converts a one-dimensional unsigned byte array to a digital signal.
	Shift	Shift the input digital signal.
	Invert	Reverse the digital data in the input digital so that 0 convert 1 (or 1 convert 0).
	GetSubset	Returns a subset of digital inputs used to plot extracted signals one by one or group on a graph.
	AppendSignals	Combining two digital signals.

	AppendSamples	Add all samples of the digital waveform src2 after the digital waveform src1.
AnalogToDigital		Convert analog signals into digital signals.
DigitalToAnalog		Convert analog signals into digital signals.

The Median Filter is a non-linear digital filtering technique, often used to remove noise from a signal. The functions of this class are listed below:

MedianFilter Class	
Process	The block uses the sliding window method to compute the moving median. This block performs median filtering on the input data over time.

The ToneAnalyzer class can be used for single-tone wave form analysis.

ToneAnalyzer Class	
SingleToneAnalysis	Calculate the parameter of single-tone waveform, a ToneInfo type result is returned by this method.
ToneInfo Class	
Frequency(double)	The frequency of input waveform
Amplitude(double)	The amplitude of input waveform
Phase(double)	The phase of input waveform

Precautions:

1. MCR(R2017a (9.2) 64-bit) and the update package should be installed before running the application that has referenced this library. The installers can be acquired in the following link: MCR installer， MCR update package.
2. FindPeaksAndValleysMatlab.dll and MWArray.dll should be copied to execution directory before execution. This operation is always done when compiling the project. While sometimes these two files are not copied to execution directory and this will lead to a FileNotFoundException. Copy FindPeaksAndValleysMatlab.dll and MWArray.dll from SeeSharpTools installer directory to the execution path will solve this problem.

2.10 SeeSharpTools.JY.Graph3D Class Library

JY.Graph3D is a custom control based on the ILNumerics class library (version 1.4) and ILNumerics.Drawing (version 0.9). The open source license is LGPL-compliant.

The class library references two kinds of chart controls such as Intensity Graph and Surface Graph.

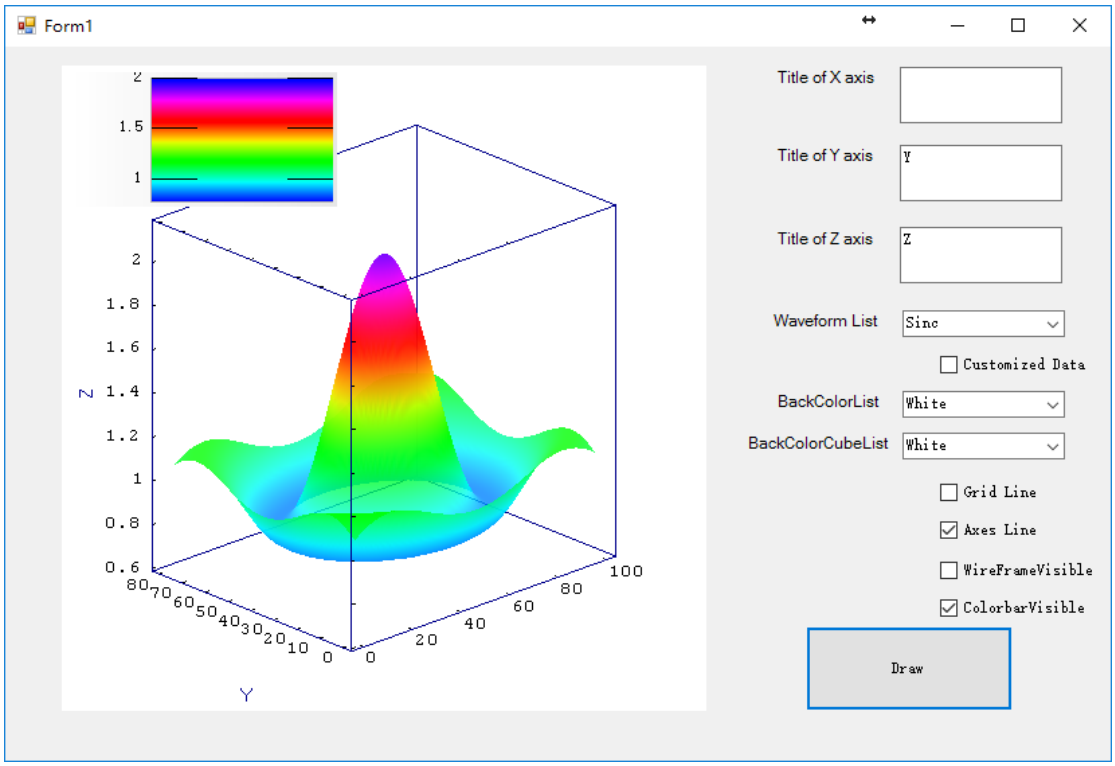
2.10.1 Surface Graph Class

Surface Graph 包含的属性以及方法如下表所示

Properties	
Property Name	**Function Summary**
AxesLineVisible	Set whether to display the axis label in SurfaceGraph, the default is True.
BackgroundColor	Set the background color outside the drawing area in SurfaceGraph. The default is White.
CubeColor	Set the color of the drawing area in SurfaceGraph. The default is White.
CubeGridVisible	Set whether to display square grid lines in SurfaceGraph. The default is false.
WireframeVisible	Set whether to display the grid on the SurfaceGraph. The default is false.
XAxisTitle	Set the dimension text of the X axis in SurfaceGraph. The default is X.
YAxisTItle	Set the dimension text of the Y axis in SurfaceGraph. The default is Y.
ZAxisTitle	Set the label text of the Z axis in SurfaceGraph. The default is Z.
BackColorOfColorBar	Set the background color of the Colorbar in SurfaceGraph. The default is White.
BorderStyleOfColorBar	Set the line type of the Colorbar in SurfaceGraph. The default is FixedSingle.
HeightofColorBar	Set the height of the Colorbar in SurfaceGraph. The default is 120.
Position	Set the position of the Colorbar in the upper left of the SurfaceGraph. The default is (5,5).
VisibleColorBar	Set whether to display ColorBar in SurfaceGraph, the default is True.
WidthofColorBar	Set the width of the Colorbar in SurfaceGraph. The default is 120.
Methods	
Method Name	**Function Summary**
Plot (double[,])	Draw a chart (giving a 2Ddouble array)
Plot (stdWfm)	Draw charts (using embedded standard arrays, Sync and Waterfall)
Refresh()	Redraw the chart

ColorBarUpdate()	Update ColorBar

After the drawing is completed, click the left mouse button to zoom in, right click to zoom out, double click to restore the default size, and drag to change the angle.



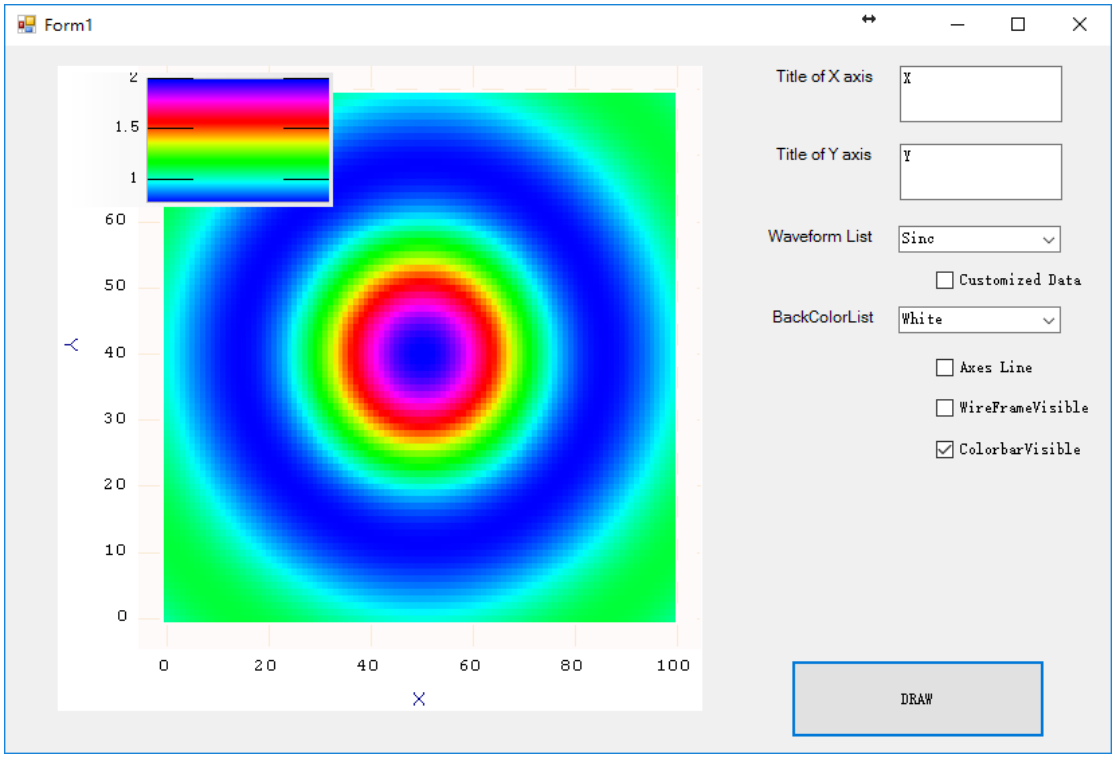
2.10.2 Intensity Graph Class

The properties and methods contained in the Intensity Graph are shown in the following table.

Properties	
Property Name	**Function Summary**
AxesLineVisible	Set whether to display the axis label in IntensityGraph. The default is True.
BackgroundColor	Set the background color outside the drawing area in IntensityGraph. The default is White.
WireframeVisible	Set whether to display the grid on the IntensityGraph. The default is false.
XAxisTitle	Set the dimension text of the X axis in IntensityGraph. The default is X.
YAxisTItle	Set the label text of the Y axis in IntensityGraph. The default is Y.
BackColorOfColorBar	Set the background color of the Colorbar in IntensityGraph. The default is White.
BorderStyleOfColorBar	Set the line type of Colorbar in IntensityGraph. The default is FixedSingle.

HeightofColorBar	Set the height of the Colorbar in the IntensityGraph. The default is 120.
Position	Set the position of the Colorbar in the upper left of the IntensityGraph. The default is (5,5).
VisibleColorBar	Set whether to display ColorBar in IntensityGraph, the default is True.
WidthofColorBar	Set the width of the Colorbar in the IntensityGraph. The default is 120.
Methods	
Method Name	**Function Summary**
Plot(double[,])	Draw a chart (giving a 2Ddouble array)
Plot(stdWfm)	Draw charts (using embedded standard arrays, Sync and Waterfall)
Refresh()	Redraw the chart
ColorBarUpdate()	Update ColorBar

After the drawing is completed, click the left mouse button to zoom in, right click to zoom out, double click to restore the default size, and drag to change the angle.



2.11 SeeSharpTools.JY.File Class Library

2.11.1 AnalogWaveformFile Class

The JY.File class library contains an AnalogWaveformFile class that provides waveform storage and reading functions. The public properties and methods are shown below.

Properties	
Property Name	**Function Summary**
FileFormat	File format, enumeration, Stream/Multi-Record.
DataType	Data format, enumeration, Int16/Int32/Float/Double.
ByteOrder	Endian, enumeration, Little Endian/Big Endian.
ArchiveInformation	For information on archiving, see the "ArchiveInformation" class description.
NumberOfChannels	Number of channels.
SampleRate	Sampling rate, S/s
DataLength	Data length, in Sample units
Channels	Sample channel information, including channel name, acquisition range, conversion factor, physical unit name and remarks, List<ChannelInfo> list, see the "ChannelInfo" class description.
DataStartTime	Data starting point time.
TimeLabels	A timestamp tag sequence that defines a set of tags to identify the time at which the specified data sample point is located. When the file format is Stream, the time can be used for long-term continuous data. When the file format is Multi-Record, it is used to identify a Record. For the starting point time information, see the "Timelabel" class description.
CustomLabels	Custom tag sequence, you can define a set of tags to save the note information, see the "CustomLabel" class description.
EnableWriteBuffering	Whether to enable the write operation cache.
EnableFastStream	Whether to enable high speed reading and writing.
Methods	
Open	Create a new file or open a file.
Close	Close the file.
Write	Write data, polymorphic methods, support multiple data types (short, int, float, double).
Read	Read data, polymorphic methods, support single channel,

	multi-channel, and multiple data types (short, int, float, double).
SetFilePosition	Read the file location in Sample.
GetFilePositon	Get the current file read location, or the length of the data that has been written, in Sample.
AddTimeLabel	Add time stamps, polymorphic methods.
GetTimeLabel	Get the specified time tag information.
AddCustomLabel	Add custom notes to support Int/Double/String.
GetCustomLabel	Get the specified custom label information.

ArchiveInformation class defines file archive information.

Properties	
Property Name	**Function Summary**
FileVersion	只读,WaveformFile 的版本
Author	可用于描述数据来源
DataLabel	可用于数据的归档标识
DataGroupID	可用于文件的关联标识
FileCreatedDate	只读,文件建立的时间
GeographicLocation	可用于描述地理位置信息
Description	可用于描述数据备注

The ChannelInfo class defines channel information.

Properties	
Property Name	**Function Summary**
Name	通道名称
RangeHigh	通道量程上限
RangeLow	通道量程下限
Offset	a of "a + b * x"
Scale	b of "a + b * x"
Unit	单位
Description	备注

The TimeLabel class defines time tag information.

Properties	
Property Name	**Function Summary**
Name	标签名.
Postion	样点位置,表示相对于起始样点的样点个数.
Time	样点时间.
Description	备注.

The CustomLabel class defines custom tag information and provides methods for

getting and modifying custom tag values.

Properties	
Property Name	**Function Summary**
Name	Tag name.
DataType	Tag data format, enumeration, Int16/Int32/Double/String.
Methods	
SetValue	Set the tag value.
GetValue	Get the tag value.

2.11.2 BinHandler Class

The BinHandler class implements the function of reading and writing data from a bin file. Currently, it supports reading and writing single or two-dimensional data of double or string type.

Methods	
Method Name	**Function Summary**
ReadData	Read data from bin file to two-dimensional string array
ReadDoubleData	Read data from bin file to 2D double array
WriteData	Write a one- or two-dimensional string or double array to a bin file. You can configure whether to append or overwrite when the file exists.

2.11.3 CsvHandler Class

CsvHandler is a static class that implements the function of reading and writing data from csv files. Currently, it supports reading and writing one-dimensional or two-dimensional data of type string, double, int, uint, short, and ushort.

Methods	
Method Name	**Function Summary**
Read<TDataType>	Read data from a csv file into a one- or two-dimensional array. The supported data types are: double/float/int/uint/short/ushort/byte.
Write<TDataType>	Write a one- or two-dimensional array to the csv file, support all types that support the ToString method, and determine that the supported data types are double/float/int/uint/short/ushort/byte.

The Read and Write methods of CsvHandler are generic methods.

The Read method supports data of type double/float/int/uint/short/ushort/byte. You need to manually specify the type of the generic type, such as the code when reading the double type: Read<double>(filepath, ...). The Read method has Multiple overloads, respectively, support reading a specified line / multiple lines starting from a line / all lines starting from a line, specifying a column / multiple specified columns, supporting input parameters containing paths and files not containing files Enter the parameter (when the path is not included, the winform file selection window will pop up), and support the configuration of the read encoding format (if not configured, the system default encoding format is used by default, most systems are unicode).

The Write method theoretically supports all types that implement the ToString method, and determines that data of type double/float/int/uint/short/ushort/byte is supported. The Write method supports writing one-dimensional and two-bit arrays, and can be configured to write the encoding format (If not configured, the default encoding format of the system is used by default, most systems are unicode), and the input parameters including the path and the input parameters not including the file are supported (the file selection window of winform will pop up when the path is not included).

Note: The interfaces of CsvHandler in 1.4.3 and earlier are marked as obsolete, and are not maintained in the later stage. It is not recommended. Applications based on the original interface can still run normally.

2.11.4 IniHandler Class

IniHandleris a static class which implements the ini file. There are 6 interfaces, the first 4 mainly used for writing and reading the ini file, and the later 2 methods used for writing and reading a single key value pair. The following are the methods:

Methods	
Method Name	**Function Summary**
ReadIniFile	Read data from the Ini file, return IniData data type
WriteIniFile	Write IniData data into file
MergeIniData	Merge two IniData data, write into first IniData
MergeToFile	Merge data in IniData with IniData in another file, write into original file
Read	Read all Section names in Ini file; Read all Keys under some Section;Read the corresponding Value to a some Key under some Section
Write	Add key value pair or modify corresponding value to a key under some Section in Ini file

There are three relevant data types in IniHandler, their usage is as follows:

Data Type

Class		Function Summary
IniData		Save class of Ini file data structures
	Sections	Collection of all Sections, use foreach to loop through
	Global	Collection of all global key value pairs
	Sections.AddSection(string sectionName)	Add blank section
	Sections.AddSection(SectionData sectionData)	Add already existing section
	Sections.ContainsSection(string sectionName)	Determines whether or not contains section with section name
	Sections.RemoveSection(string sectionName)	Remove certain section
	Sections[string sectionName]	Get SectionData of Section with sectionName
Section		Save class of single section data
	SectionName	Get or Configurate section name
	Keys	Save Collection of Keys to Values. Use foreach to loop through.
	Keys.AddKey(string key, string value)	Add a key value pair
	Keys.RemoveKey(string key)	Delete a key value pair
	Keys.ContainsKey(string key)	Whether or not contains a key value pair
	Keys[string keyName]	Get Value corresponding to key. Use Key[key]=value to modify name of some key value pair.
KeyData		Save class of KeyData
	KeyName	Name of key
	Value	Name of value

2.12 SeeSharpTools.JY.Report Class Library

JY.DSP.Report contains 2 classes：WordReport，ExcelReport and Logger, providing word report creations, Excel report creations and Log functions:

2.12.1 WordReport Class

WordReport Class	
Property Name	**Function Summary**
DefaultFont	Default Word text format, including font, color, bold, italic, underline
Method Name	**Function Summary**
Close	Close word engine
SaveAs	Save as
Show	Show Word program
Hide	Hide Word Program in background
WriteTextToDoc	Write text to document
WriteTableToDoc	Write table to document
InsertGraph	Insert graph
InsertPicture	Insert picture
ExecuteMacroFromScript	Execute macro from script

2.12.2 ExcelReport Class

ExcelReport Class	
Property Name	**Function Summary**
DefaultFont	Defualt Excel text format, including font, color, bold, italic, underline
RowCount	Get Excel report row count
ColumnCount	Get Excel report colun count
Method Name	**Function Summary**
Hide	Hide Excel program in background
Show	Show Excel program
SaveAs	Save as
Close	Close Excel engine
RCToString	Convert Excel row column numbers into row column strings
StringToRC	Convert row column strings into row column numbers
LetterToInt	Convert letter into Excel integer

zsIntToLetter	Convert integer into Excel letters
WriteTextToReport	Write text to report
WriteTableToReport	Write table to report
WriteListToReport	Write list to report
InsertGraph	Insert graph
WriteArrayToReport	Write array to report
AppendRow	Append row
AppendColumn	Append column
ReadSingleCell	Read single cell
ReadConsecutiveCells	Read consecutive cells
ReadRegionCells	Read region cells
ReadCurrentSheet	Read current sheet
ReadSheet	Read sheet
ReadAllSheets	Read all sheets
ExecuteMacroFromScript	Execute macro from script

2.12.3 Logger Class

Logger Class is a thread-safe static Logger class. The current version supports writing logs into a single file or a certain directory. Properties and methods of the Logger class are as follows:

Logger Class	
Static Property Name	**Function Description**
LogLevel (LogLevel Enumeration)	Configure and get the log level of the current log, including increasing log levels Trace/ Debug/ Info/ Warn/ Error/ Fata. Log info lower than LogLevel will not be written to the log.
LogConfig (LogConfig Class)	LogConfig Class, save logger's configuration data.
Enabled(bool)	Configure enable/disable logging.
Static Method Name	**Function Summary**
Initialize	Initialize log, can input log file path (single file log) and LogConfig
Print	Print logs at the specified logging level, multiple overloads
Trace	Write to log at level Tracing
Debug	Write to log at level Debug
Info	Write to log at level Info
Warn	Write to log at level Warn
Error	Write to log at level Error

Fatal	Write to log at level Fatal
StackTrace	Write to current stack trace at assigned LogLevel, default debug LogLevel
ThreadInfo	Write to current thread trace at assigned LogLevel, default debug LogLevel
Close	Close the current log session
LogConfig Class	
Property Name	**Function Description**
Type(LogTypeEnumeration)	Log type currently supports FileLog only, which is the log for the file
FileLog (FileLogConfig Class)	Configuration class for the file log
LogFormat(string)	Format of log printing, {0} is LogLevel, {1} is time stamp, {2} is log info, default is "[{0}] [{1}] {2}", print example as follows： `[Warn] [2018-06-06 11:55:02:685] This is a test.`
ExceptionFormat(string)	Exception format of log printing, {0} is LogLevel, {1} is time stamp, {2} is exception type, {3} is exception message
StackTraceFormat(string)	Stack trace format when log printing, {0} is printing info
ThreadInfoFormat(string)	Thread trace format when log printing, {0} is thread ID, {1} is thread name
TimeStampFormat(string)	Time stamp format,default is：yyyy-MM-dd hh:mm:ss:fff
Header(string)	Log header
FileLogConfig Class	
Property Name	**Function Description**
Extension(string)	File extension, default is log
LogMode(FileLogMode Enumeration)	Log mode, includes single file log and directory log
Path(string)	Log file path or file directory
LogNameFormat(string)	The time format of the LogName under directory log mode, default is : yyyy-MM-dd hh-mm-ss
MaxLogSize(long)	Max log size of file, default is 100M. When exceed value: clear all written data in single file log;create new log file in log directory to write to
Encode(Encoding Class)	Encoding format, default is Unicode
Flush(FlushType Enumeration)	File flush mode: SyncFlush, AsyncFlush

2.13 SeeSharpTools.JY.ThreadSafeQueue Class Library

The class library includes Thread safe queue components.

2.13.1 ThreadSafeQueue Class

ThreadSafeQueue is a thread-safe queue based on Queue development. It saves object references (not copying values). All data are objects saved in the queue, and remain objects leaving the queue. If there is the need to read and write non-object data, must convert first. Queue expands at full capacity.

FileLogConfig Class	
Method Name	**Function Summary**
Clear	Clear queue
Enqueue	Enqueue data (object references)
Dequeue	Dequeue data

2.13.2 CircularQueueClass

CircularQueue is a thread-safe queue based on array development, implemented with generic types, and saves non-String-type objects. Passes value when saving data. Queue size is fixed when constructed to prevent overflow. Size can also be initialized to a larger value according to amount of data written. In the meantime, the object implements IList interface, allowing random reads on data saved in the queue.

CircularQueue Class	
Property Name	**Function Summary**
Count(int)	Get the count of data in queue
Capacity(int)	Get the maximum capacity of the queue, parameter given in constructor
AutoLock(bool)	Whether to use AutoLock(Lock that automatically operates), default is true
BlockWait(bool)	Whether to block reading thread if not enough data when reading, default is true
Method Name	**Function Summary**
Clear	Clear Queue

Enqueue	Enqueue data. Throws exception when time out occurs for getting lock
TryEnqueue	Enqueue data. Return false when time out occurs for getting lock, else return true
Dequeue	Deque data. Throws exception when time out occurs for getting lock
TryDequeue	Deque data. Return false when time out occurs for getting lock, else return true
Enter	Gets operating queue's lock under OCC mode. Throws exception when time out occurs for getting lock
TryEnter	Gets operating queue's lock under OCC mode. Return false when time out occurs for getting lock,
Leave	Releases operating queue's lock under OCC mode.

2.14 SeeSharpTools.JY.Sensors Class Library

SeeSharpTools.JY.Sensors class library includes 6 sensor type calculations, enabling the sampled values to be converted into physical quantities of the sensor.

2.14.1 CustomScaling Class

CustomScaling Class	
Method Name	**Function Summary**
Convert	Customize Func<double,double> function

2.14.2 DisplacementSensor Class

DisplacementSensor Class	
Method Name	**Function Summary**
Convert	Displacement Sensor

2.14.3 LoadCell Class

LoadCell Class	
Method Name	**Function Summary**
Convert	Load Cell

2.14.4 RTD Class

RTD Class	
Method Name	**Function Summary**
Convert	RTD (resistance temperature detectors)

2.14.5 Thermistor Class

Thermistor Class	
Method Name	**Function Summary**
Convert	Thermistor

2.14.6 Thermocouple Class

Thermocouple Class	
Method Name	**Function Summary**
Convert	Thermocouples (supports types B,E,J,K,N,R,S,T)

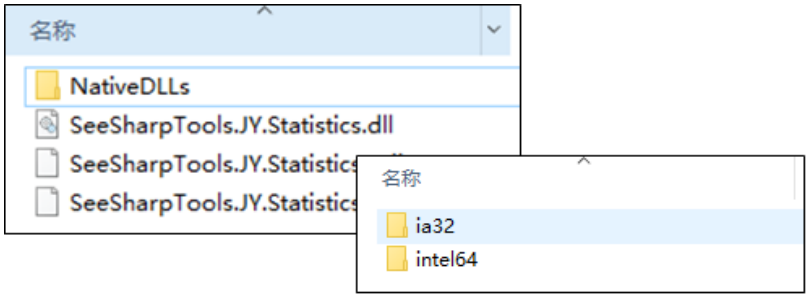
2.15 SeeSharpTools.JY.Statistics Class Library

JY. Statistics dll includes 9 common statistical functions，integrating Math.Net libraries and high-performance Intel IPP libraries optimized for Intel CPU (Switch by easily modifying properties to calculate core categories).

Download the Intel IPP dll online and install it if needed. According to your running platform, put the dll in either the ia32 folder or intel64 folder under the same directory as SeeSharpTools.JY.Statistics, as shown below.

..\NativeDLLs\ia32\ →32 bit dll

..\NativeDLLs\intel64 → 64bit dll



Engine Class	
Property	**Function Summary**
Provider	Use Math.NET or IntelIPP according to the running engine
Statistics Class	
Method Name	**Function Summary**
Histogram	Plot array histogram
Kurtosis	Calculate Kurtosis value
Maximum	Calculate Maximum value
Mean	Calculate mean value
Minimum	Calculate minimum value
RMS	Calculate RMS value
Skewness	Calculate Skewness value
StandardDeviation	Calculate Standard Deviation
Variance	Calculate Variance
CalculateCapability	Calculate the SPC result of given samples. The results are stored in an instance of CapabilityResult Class.
CapabilityResult 类	
属性名	功能描述
Mu	The mean value of samples

Sigma	The standard deviation of samples
P	The probability of samples that within the limit
Pl	The probability of samples that below than the low limit
Pu	The probability of samples that higher than the high limit
Cp	(High - Low)/(6*Sigma)
Cpl	(Mu - L)./(3*sigma)
Cpu	(U - Mu)/(3*Sigma)
Cpk	min(Cpl, Cpu)
Ca	(Sigma-C)/(Et*0.5)

Precautions:

1. MCR(R2017a (9.2) 64-bit) and the update package should be installed before running the application that has referenced this library. The installers can be acquired in the following link: MCR installer， MCR update package.
2. StatisticsMatlab.dll and MWArray.dll should be copied to execution directory before execution. This operation is always done when compiling the project. While sometimes these two files are not copied to execution directory and this will lead to a FileNotFoundException. Copy StatisticsMatlab.dll and MWArray.dll from SeeSharpTools installer directory to the execution path will solve this problem.

2.16 SeeSharpTools.JY.TCP Class Library

JY.TCP dll includes two classes：JYTCPServer and JYTCPClient。

JY.TCP dll provides two data transfer modes: string transfer mode and data transfer mode. Mode chosen at object creation. Server supports multiple client connections.

JYTCPServer Class	
Property Name	**Function Summary**
ConnectedClients	The information of connected clients.
BufferSize	Buffer Size
LocalIP	The IP address that the server binded.
Method Name	**Function Summary**
Start	Start
Stop	Stop
ReadDataStream	Read data from specified client, the one-dimensional array or two-dimensional array of double/float type is supported.
ReadString	Read string from specified client.

SendDataStream	Send data to specified client, the one-dimensional array or two-dimensional array of double/float type is supported.
SendString	Send string to specified client.
Event	**Function Summary**
ClientDisconnected	Event triggered when client disconnects
ClientConnected	Event triggered when client connects

JYTCPClient Class	
Property Name	**Function Summary**
AvailableSamples	Returns sample count in object cache
BufferSize	Buffer Size
Connected	Determine if connection is successful
Method Name	**Function Summary**
Connect	Start connection
DisConnect	Terminate connection
ReadDataStream	Read data from server, the one-dimensional array or two-dimensional array of double/float type is supported.
ReadString	Read string from server.
SendDataStream	Send data to server, the one-dimensional array or two-dimensional array of double/float type is supported.
SendString	Send string to server.
Event	**Function Summary**
ServerDisconnected	Event triggered when disconnected from server

2.17 SeeSharpTools.JY.Mathemetics Class Library

JY.Mathematics dll provides various mathematical calculation static libraries including array operations/calculations, statistics, calculus, curve fitting, interpolation algorithms, linear algebra, e.t.c.

ArrayArithmetic Class	
Method Name	Function Summary
Absolute	Absolute Value
Acos	Inverse Cosine
Add	Array Addition
Asin	Inverse Sin
Atan	Inverse Tangent
Cos	Cosine
Divide	Array Division
Exp	Exponential
FindMaxMin	Find max/min
Initialize	Initialize
Ln	Natural Log
Log	Log
Multiply	Array Multiplication
Pow	Power
Product	Array Multiplication then Addition
Sin	Sin
Sqrt	Square root
Substract	Array subtraction
Sum	Sum
Tan	Tangent

ArrayOperation Class	
Method Name	Function Summary
Concatenate	Concatenate
ConvertTo	ConvertTo
Copy	Copy
Delete	Delete
GetSubset	Get Subset
Insert	Insert
Inverse	Inverse
ReplaceSubset	Replace Subset

Transpose	Transpose

Statistics Class	
Method Name	Funciton Summary
Histogram	Histogram
Kurtosis	Kurtosis
Mean	Mean
Median	Median
Percentile	Percentile
RMS	RMS
Skewness	Skewness
StandardDeviation	StandardDeviation
Variance	Variance

LinearAlgebra Class	
Method Name	Function Summary
Dot	Dot
Norm	Norm

Interpolation Class	
Method Name	Function Summary
Interpolate_CubicSpline	Interpolate_CubicSpline
Interpolate_Linear	Interpolate_Linear
Interpolate_LogLinear	Interpolate_LogLinear
Interpolate_Polynomial	Interpolate_Polynomial
Interpolate_Step	Interpolate_Step

Fitting Class	
Method Name	Function Summary
ExponentialFitting	Exponential fitting
LinearFitting	Linear fitting
TransposePolynomialFitting	Transpose polynomial fitting

Calculus Class	
Method Name	Function Summary
Derivative_2ndOrderCentral	2nd order derivative
Derivative_4thOrderCentral	4th order derivative
Derivative_Backward	Backward derivative
Derivative_Forward	Forward derivative

Integral_Bode	Bode integral
Integral_Simpsons	Simpson integral
Integral_Simpsons38	Simpson3/8 intergal
Integral_Trapezodial	Trapezodial integral

Vector 类	
方法名	功能概要
ArrayAbs	Array Absolute
ArrayAdd	Array Addition
ArrayAscend	Array Ascend
ArrayAscendIndex	Array Ascend, return index
ArrayConvert	Array Convert
ArrayCopy	Array Copy
ArrayDescend	Array Descend
ArrayDescendIndex	Array Descend, return index
ArrayDivision	Array Division, return index.
ArrayDotProduct	Array Dot Product
ArrayDotProductConj	Array Dot Product Conj
ArrayEqual	Array Equal
ArrayExp	Array Exponent
ArrayExp10	Array Exponent 10
ArrayExp2	Array Exponent 2
ArrayFindMax	Array Find Max
ArrayFindMin	Array Find Min
ArrayFindMaxMin	Array Find Max and Min
ArrayLn	Array Ln
ArrayLog10	Array Log 10
ArrayLog2	Array Log 2
ArrayMax	Take max value as array elements.
ArrayMean	Take mean value as array elements
ArrayMin	Take min value as array elements
ArrayMulti	Take product value as array elements
ArrayPhaseMean	Taske mean for phase array
ArrayPhaseMeanWeight	Take mean for phase array with weight
ArrayReserve	Reverse array

ArrayRoot	Take root for array elements
ArrayRotate	Rotate array
ArrayScale	Scale array by constant value
ArraySquare	Take square for array element
ArraySub	Array Subtraction
ArraySum	Array Summation
ArrayTranspose	Transpose Array

Specific functions under JY.Mathematics support the accelerated computing power based on Intel technology. The operation methods and support list are as follows:

【Accelerate with Intel IPP】

1. Install **intel IPP** software package (download from Intel's website)
2. Add a new "IntelIPP" folder to the build directory.
3. Copy the .dll from the intel package to the "IntelIPP" directory. Copy the x86 version .dll for x86 programs, the x64 version for x64 programs.

   x86 Path:

   C:\Program Files (x86)\IntelSWTools\compilers_and_libraries_2017.4.210\windows\redist\ia32\ipp

   x64Path:

   C:\Program Files (x86)\IntelSWTools\compilers_and_libraries_2017.4.210\windows\redist\intel64\ipp
4. Add the following line to code:

   Engine.Provider = ProviderEngine.IntelIPP
5. Supporting list as follows:

ArrayArithmetics	ArrayOperation	Statistics	LinearAlgebra
Absolute	Inverse	Mean	Dot
Acos		StandardDeviation	
Add			
Asin			
Atan			
Cos			
Divide			
Exp			
FindMaxMin			
Initialize			
Ln			
Log			

Multiply			
Pow			
Product			
Sin			
Sqrt			
Substract			
Sum			
Tan			

【Accelerate with Intel MKL】

1. Install the MKL runtime package (Download from jytek.com)
2. Add the following line to code:

   Engine.Provider = ProviderEngine.IntelMKL

ArrayArithmetics	Statistics	LinearAlgebra
Absolute	Kurtosis	Dot
Acos	Skewness	
Add		
Asin		
Atan		
Cos		
Exp		
FindMaxMin		
Ln		
Log		
Multiply		
Pow		
Sin		
Sqrt		
Substract		
Sum		
Tan		

2.18 SeeSharpTools.JY.DSP.Measurements Class Library

JY.DSP.Measurements provides class library for various measurements. Currently, it provide static library for square wave measurement.

SquarewaveMeasurements Class	
Method Name	Function Summary
AmplitudeLevelAnalysis	Square wave amplitude analysis: analyze the amplitude information of the input waveform. Returns: amplitude and high-low level values according to the high-low level state configuration (by peak or histogram).
PeriodAnalysis	Square wave period analysis: analyze the periodic characteristics of square wave according to the input waveform and sampling interval, percentage level configuration, pulse polarity and reference level configuration (percentage or absolute value, high level, middle level and low level). Returns: cycle information (number of cycles, maximum value, minimum value and average value), pulse (number of pulses, maximum value, minimum value and average value), duty cycle (number of duty cycles, maximum value, minimum value and average value).
PulseMeasurement	Square wave pulse measurement: analyze the pulse information of square wave according to the input waveform, sampling interval, percentage level configuration, pulse polarity, reference level configuration (percentage or absolute value, high level, middle level and low level) and pulse number. Returns: the period, pulse width, duty cycle and measurement information(including the time of the pulse center and the reference level setting used for measurement) of the specify pulse.
TransitionMeasurement	Square wave transition edge measurement: analyze the transition edge information of square wave according to the input waveform, sampling interval, percentage level configuration, edge direction, reference level configuration (percentage or absolute value, high level, middle level and low level) and edge number. Returns : slope, rising/falling time, edge overshoot and undershoot information and measurement information (including edge start and end time, reference level setting used for measuring edge) of specified edge.

3. FAQ

For additional help, please contact Shanghai JYTEK INC. Our email is：mailto:support@jytek.com