
Shanghai Geshe Information Technology Co., Ltd.

**Geshe MC Master
User Manual**

Version 1.0

Contents

1. Introduction	3
1.1 About	3
1.2 Functional Characteristics	3
1.3 System Requirements	9
1.4 Product Licensing	10
1.5 Product Support	10
2. Quick Start	11
2.1 Installation	11
2.2 Login	12
2.3 User Interface	13
2.3.1 Main Window	13
2.3.2 Application Menu	13
2.3.3 Toolbar	15
2.3.4 Editor Area	16
2.4 Create a Data Acquisition and Monitoring Project	19
2.4.1 Step 1 – New Project	19
2.4.2 Step 2 – Add Serial Port Device	19
2.4.3 Step 3 – Add Sequence	21
2.4.4 Step 4 – Add Variants	24
2.4.5 Step 5 – Use Scripts to Associate Sequence Data with Variables	26
2.4.6 Step 6 – Add Schema	29
3. FAQ	33

1. Introduction

1.1 About

Geshe MC Master is an IDE for measurement and control software based on modular technology, which can help users develop automated test and control software system quickly. It can help enterprises to unify and standardize the development and management of test and control software, and reduce the development cost, learning cost and maintenance cost of test and control software.

Geshe MC Master provides a unified environment and interface for all different applications of automated testing and control, and provides a flexible and powerful framework for the development, management and execution of measurement and control systems, thus effectively solving the problems in four key areas:

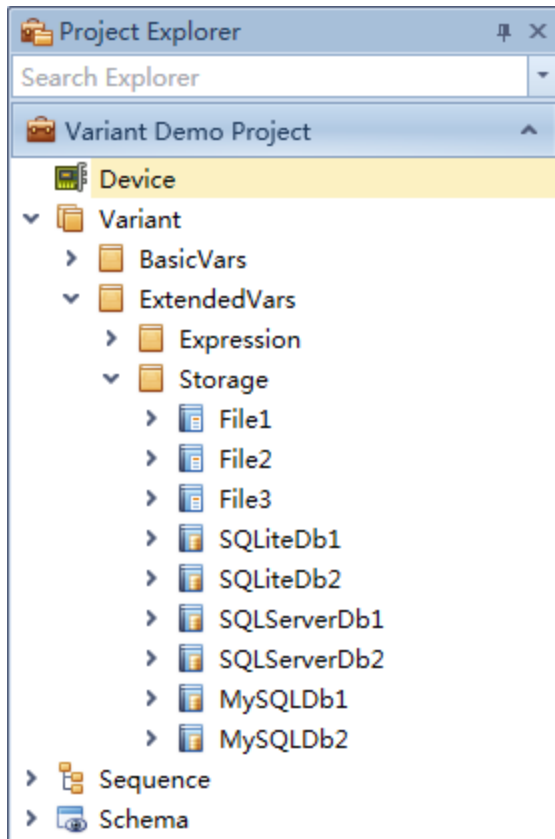
- Simplifying and accelerating the development of complex sequences
- Improving the reusability and maintainability of code and control program
- Improving the expansibility of measurement and control system
- Improving the performance of measurement and control system

For any measurement and control project that needs to accelerate development, code reuse, performance improvement and automation, such as data acquisition and monitoring system, design verification system, hardware testing system, chip testing system, etc., Geshe MC Master is indispensable.

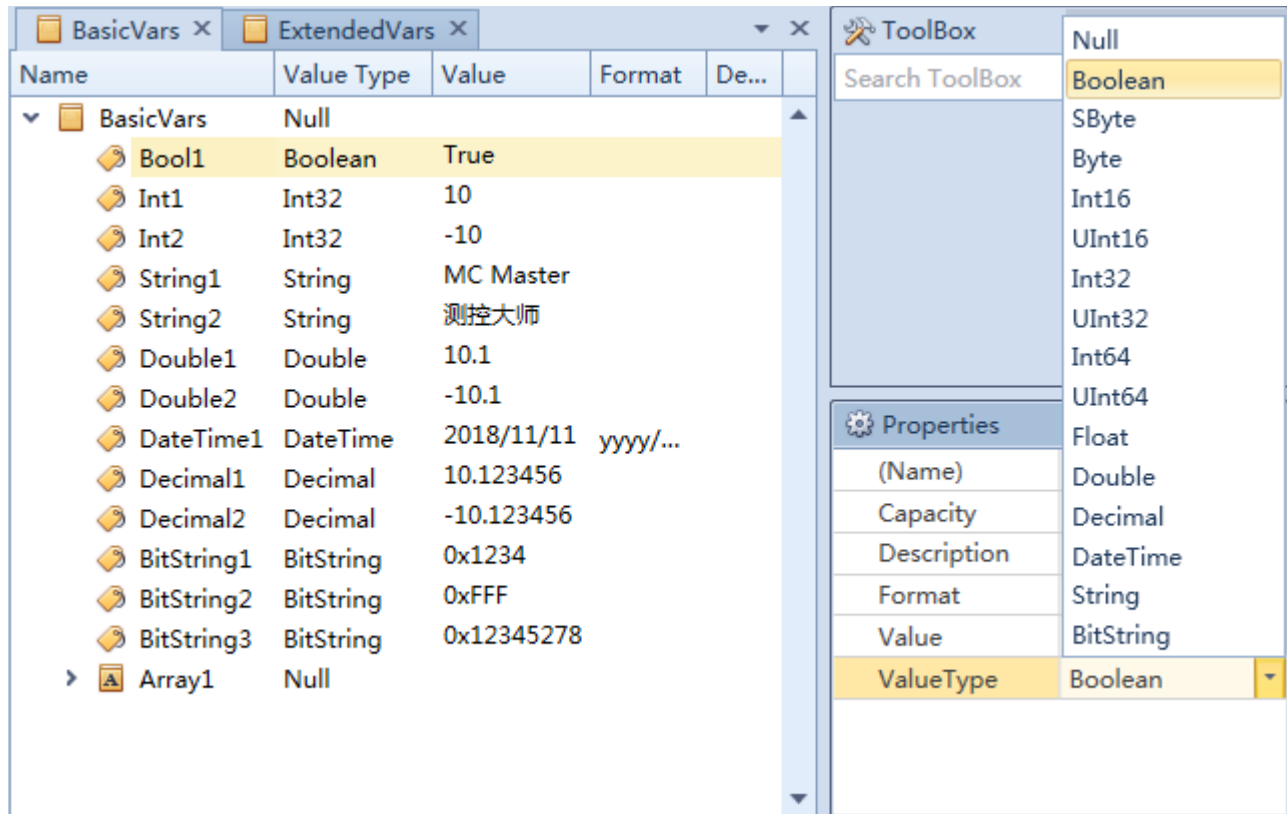
1.2 Functional Characteristics

1) Custom Device Interface

With the help of the device adapter built in the software, users can create any combination of devices and interfaces, and can communicate with different devices and interfaces at the same time to meet the needs of various measurement and control connections.



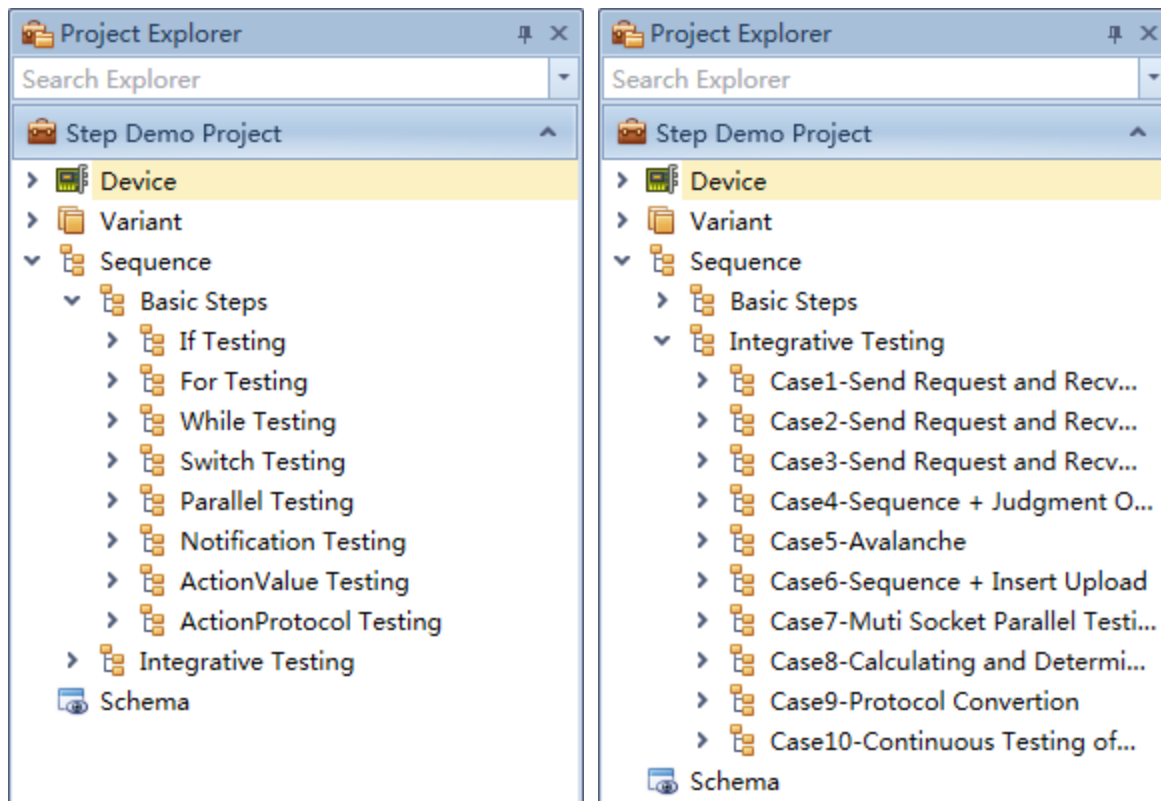
Software built-in variable editor, users can easily create and edit various types of variables.



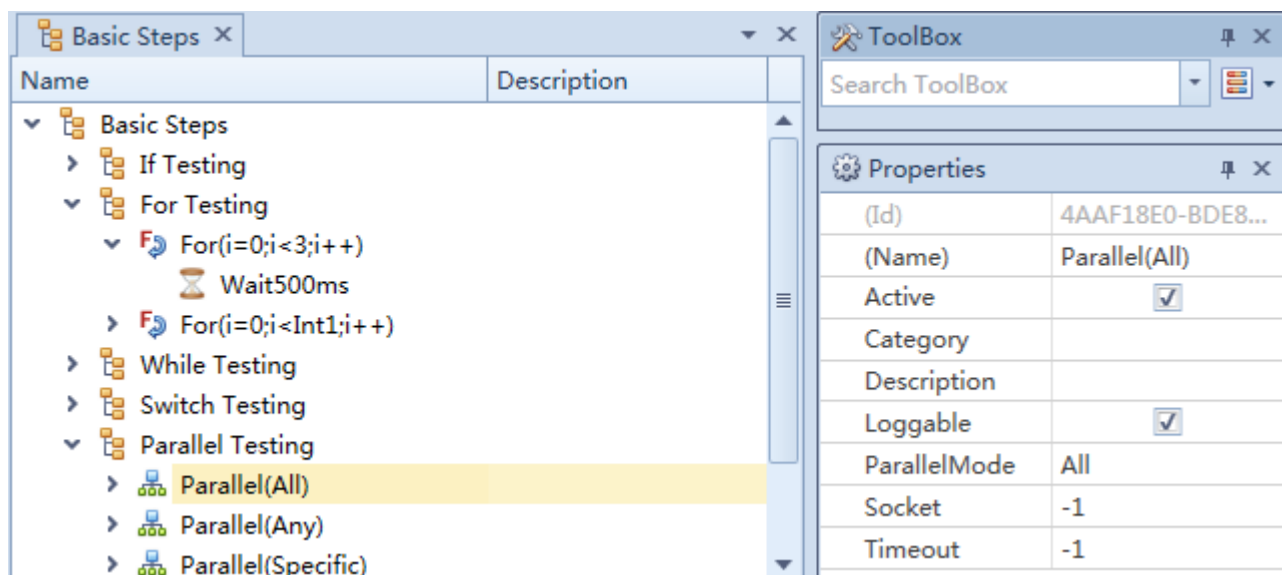
3) Custom Execution Sequence

With the help of the software built-in sequence adapter, users can create execution sequence, realize the execution process of arbitrary logic, and meet the needs of various measurement and control automation.

- Support the creation of execution sequence, realize the execution process of arbitrary logic, and meet all kinds of measurement and control execution requirements;
- Support process control, such as branch statement (If, Switch), loop statement (For, While), parallel statement (Parallel);
- Support synchronous control, such as Wait, Notification;
- Support numerical type action steps (Value) and protocol type action steps (Protocol);
- Support sequence nesting and complex hierarchy;
- Support scripts, which can seamlessly call .Net Framework class libraries and call third-party managed libraries to implement the execution logic;
- Support protocol templates and step templates.

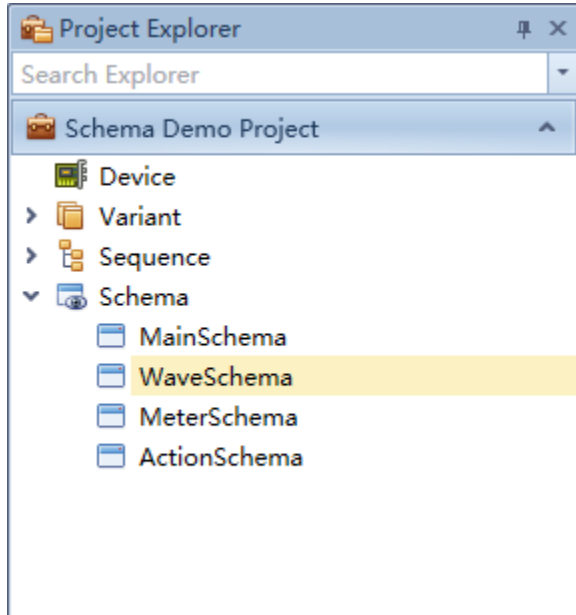


The software has a built-in sequence editor. Users can easily create and edit various types of sequence steps.



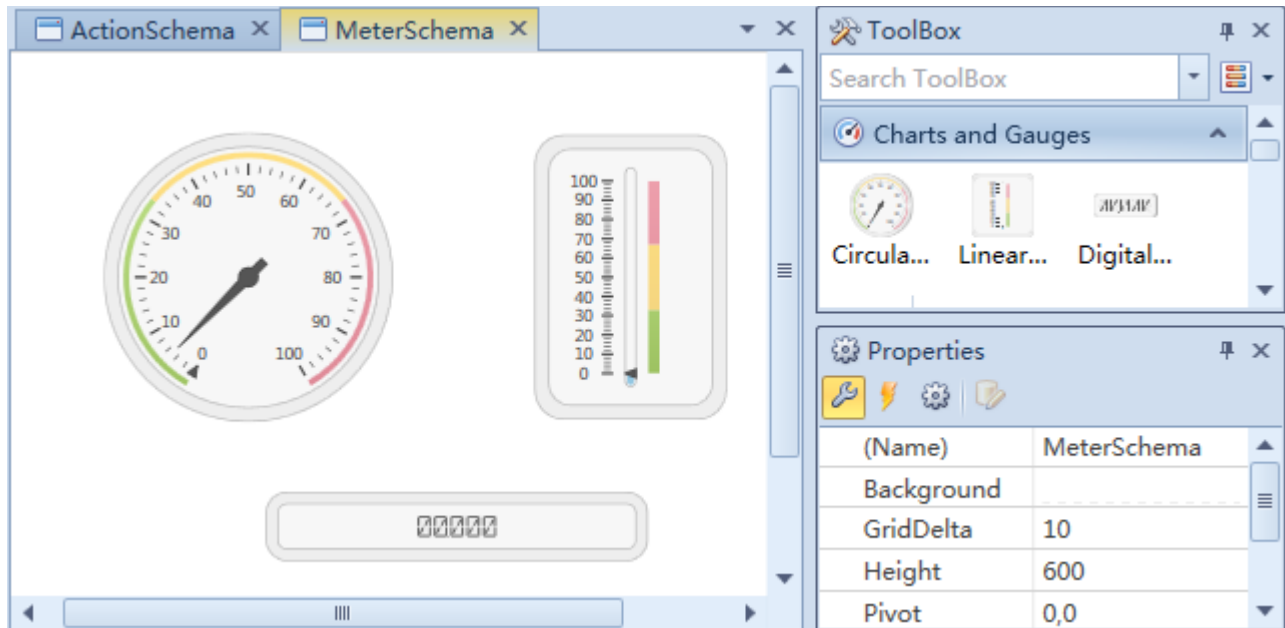
4) Custom User Interface

With the help of the built-in screen adapter in the software, users can create pictures, use the controls and shape templates of the picture toolbox to realize arbitrary user interface and meet the needs of various measurement and control interfaces.



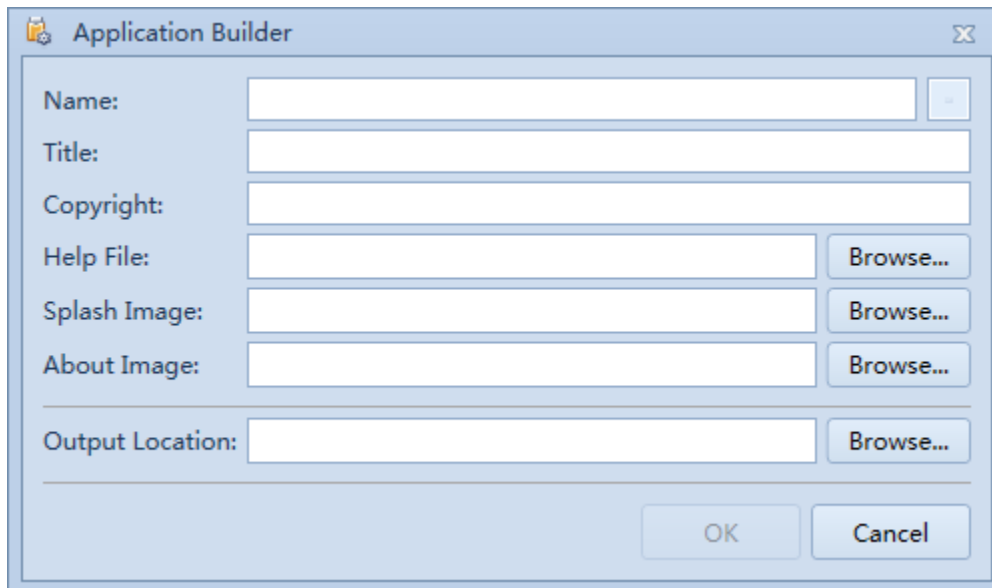
Software built-in schema editor, users can easily create and edit the schema, realize the UI logic.

- Support the creation of user interface, using toolbox controls and shape templates, to achieve arbitrary user interface and to meet the needs of various measurement and control interfaces;
- Support property data binding, which can establish the relationship between the property and variable of UI elements;
- Support event scripts, which can seamlessly call .Net Framework class libraries and call third-party managed libraries to implement the UI logic;
- Support dynamic action, which can establish the relationship between UI elements action (movement, rotation, size) and variables;
- Support control templates and shape templates.



4) Build Application

Users can create runtime application programs after the completion of the development of the measurement and control program. They can customize the software name, title, copyright, help file, startup image, about image and other software information to meet the deployment requirements of various measurement and control programs.



1.3 System Requirements

Supported Operating System:

- Windows Vista SP2 (x86 和 x64)
- Windows 7 SP1 (x86 和 x64)
- Windows 8 (x86 和 x64)

- Windows 8.1 (x86 和 x64)
- Windows Server 2008 SP2 (x86 和 x64)
- Windows Server 2008 R2 SP1 (x64)
- Windows Server 2012 (x64)
- Windows Server 2012 R2 (x64)
- Windows 10 (x86 和 x64)

Hardware Requirements:

- Recommended minimum requirements: 1 GHz or faster, 1 GB RAM or larger
- Minimum disk space: 200 MB

Essential Components:

- Microsoft .NET Framework 4.6

1.4 Product Licensing

This product is licensed by hardware encryption lock.

Permission Rules:

- An encryption lock can only authorize one computer at a time.
- Authorized computers can run multiple instances of the product at the same time.
- Encryption locks do not support virtual machine authorization.

1.5 Product Support

If you encounter problems in using this software or want to get product support information, you can contact us through our website, e-mail, etc.

- Company website:www.geshe.com
- E-mail:support@geshe.com
- QQ: 979464

2. Quick Start

2.1 Installation

Welcome to install Geshe MC Master!

Step 1 - Ensure computer support for Geshe MC Master

Before starting installation, please check the system requirements in Section 1.3 to see if the computer system supports Geshe MC Master.

Step 2 - Download Geshe MC Master

Next, download the installation file of Geshe MC Master from the official website www.geshe.com.

Step 3 - Execute the installation file

Next, run Geshe MC Master Installation File for installation.

Installer installs the following programs in turn:

- Microsoft .Net Framework 4.6 (Skips if installed)
- Geshe MC Master

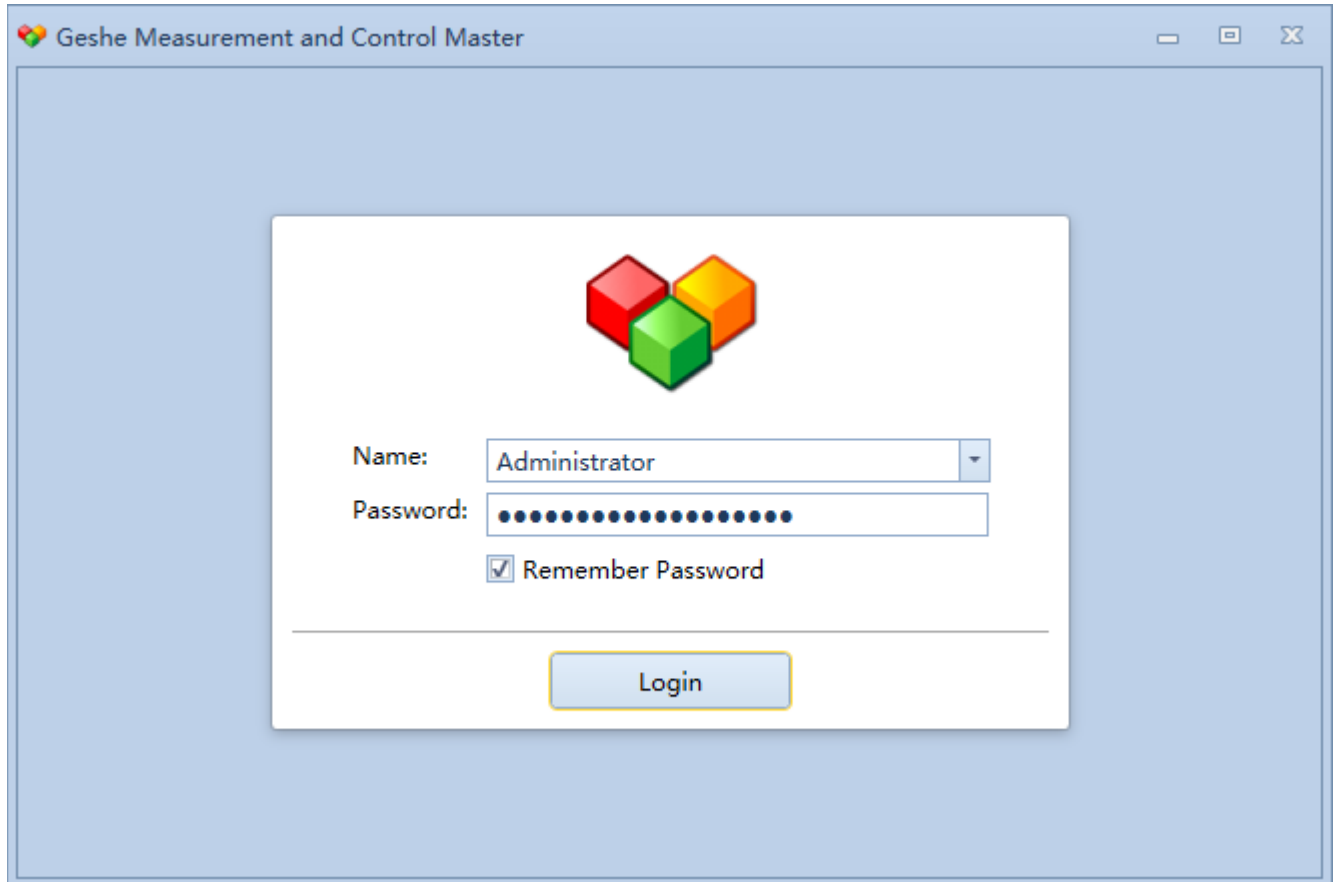


Step 4 - Start using

After the installation of Geshe MC Master, click the "Start to Use" button and start using Gussie Master to develop.

2.2 Login

Before using Geshe MC Master, you need to log in to the account. The login interface after the software starts is shown in the following figure.



The initial account list built in the software is as follows. Different accounts have different operation rights.

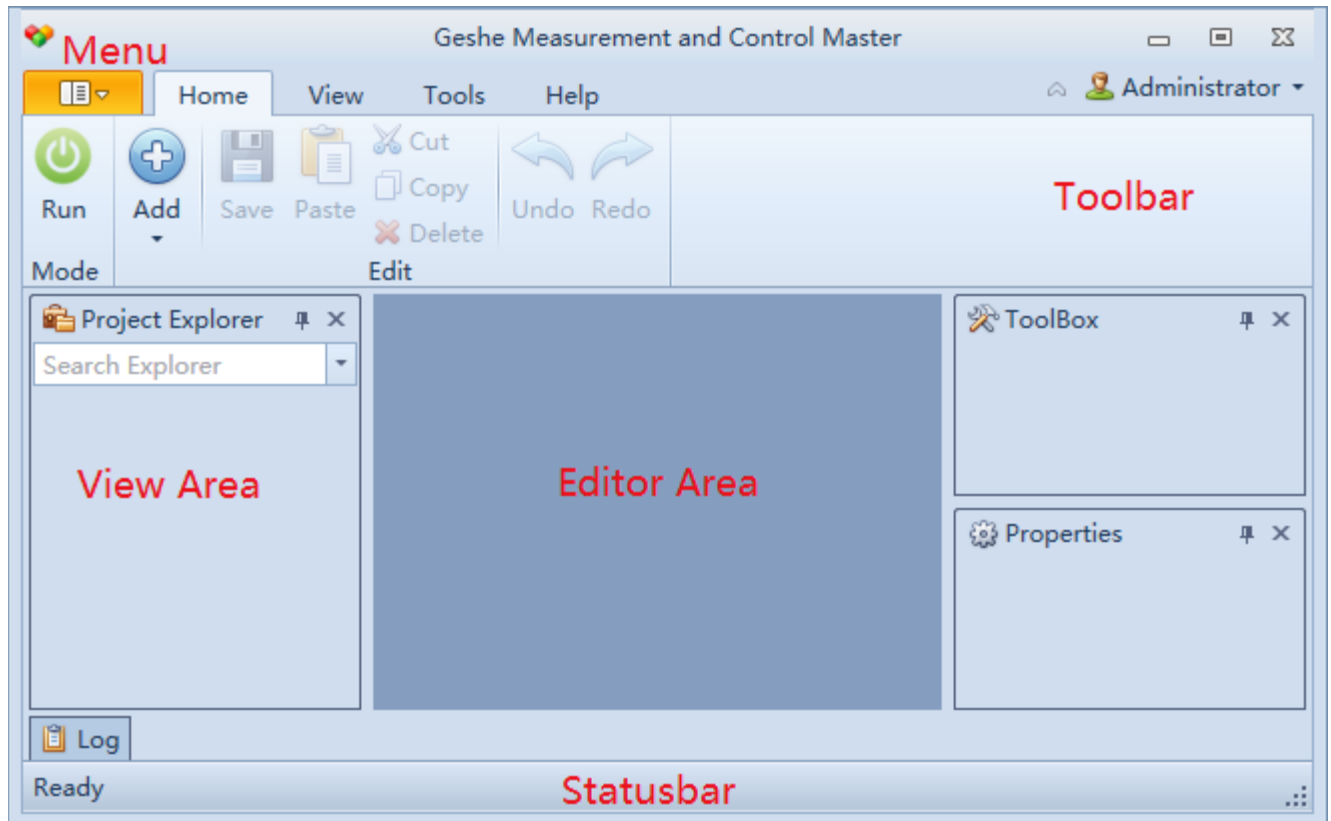
User Name	Password	Description
Administrator	(None)	Administrator, with all rights.
Developer	(None)	Developer, with all rights except "user management".
Operator	(None)	Operator, only project operation permission.

Note: The letters of user names are case sensitive, for example, the first letter of Administrator is capital A.

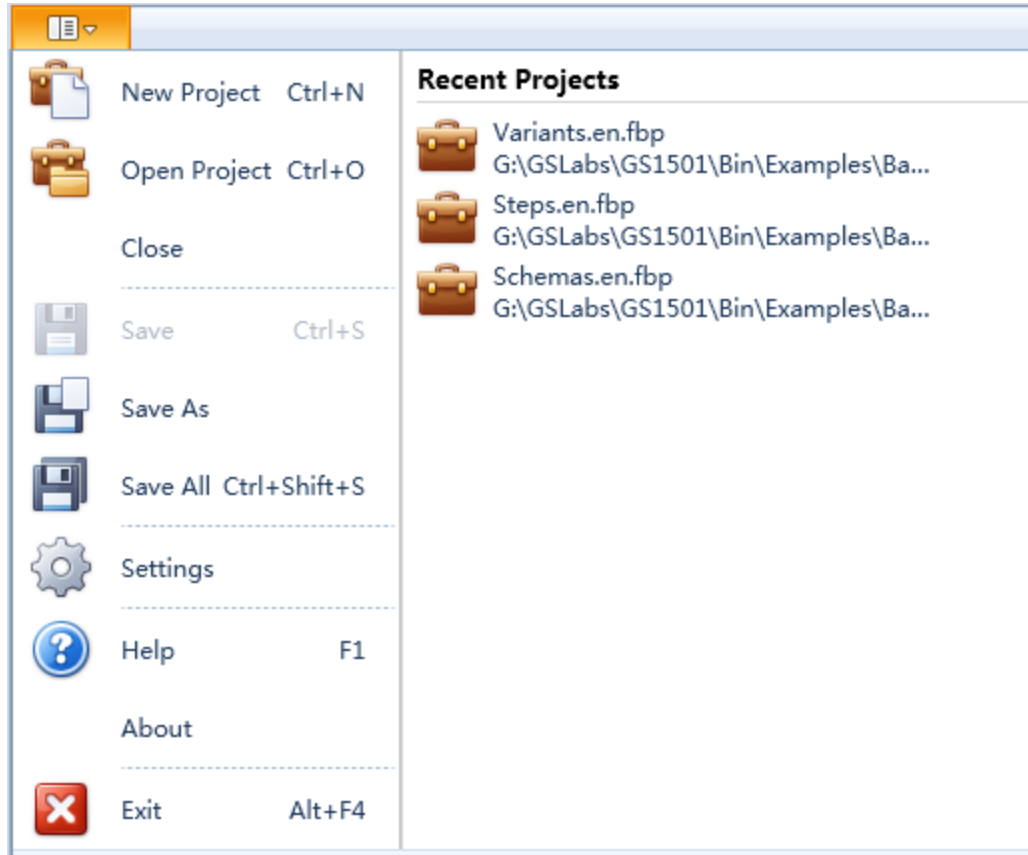
2.3 User Interface

2.3.1 Main Window

The main interface of Geshe MC Master is shown in the following figure. The top left corner is the "menu" button, which can pop up the application menu, and the next is the toolbar, view area, edit area and status bar.



2.3.2 Application Menu



Menu Item	Shortcut Key	Function
New Project	Ctrl+N	Create a new project and establish the project environment.
Open Project	Ctrl+O	Open a project and establish the project environment.
Close		Close the currently active project.
Save	Ctrl+S	Save the currently active item.
Save As		Save the currently activated project to the specified path.
Save All	Ctrl+Shift+S	Save all currently open items.
Settings		Configure system.
Help	F1	User help.
About		Display the copyright, version and registration information of the software.
Exit	Alt+F4	Exit the system.

2.3.3 Toolbar

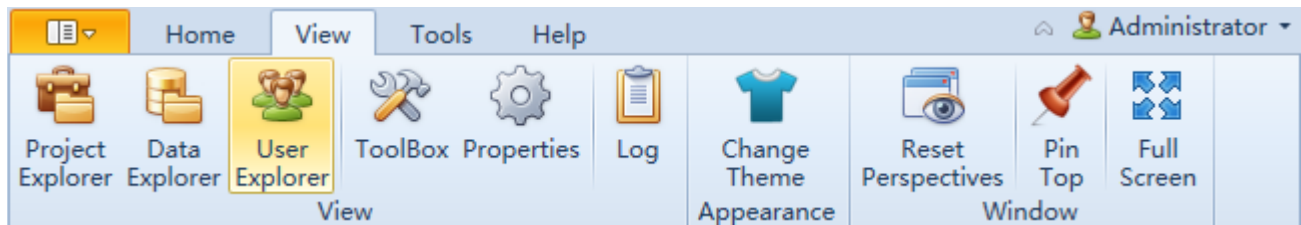
Home

The home toolbar is the main toolbar for user operation, which contains the basic command button and the command button currently activating the editor.



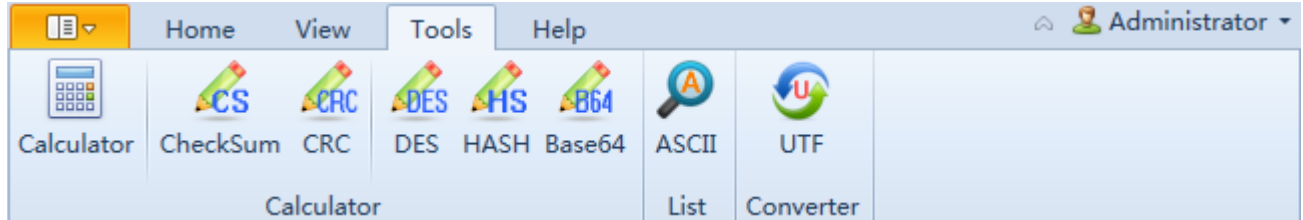
Command	Function
Run/Design	Switch to run state or design state.
Add	The submenu contains entries that can be added to the current edit status.
Collapse toolbar (top right button 1)	Display or fold toolbar.
Login user (second button in upper right corner)	Menu for login user.

View



Command	Function
Project Explorer	Open Project Explorer
Data Explorer	Open Data Explorer
User Explorer	Open User Explorer
Toolbox	Open Toolbox
Properties	Open Properties View
Log	Open Log View
Change Theme	Change Theme
Reset perspectives	Reset window arrangement as defaults
Pin Top	Pin MainWindow Top
Full Screen	Full screen display

Tools

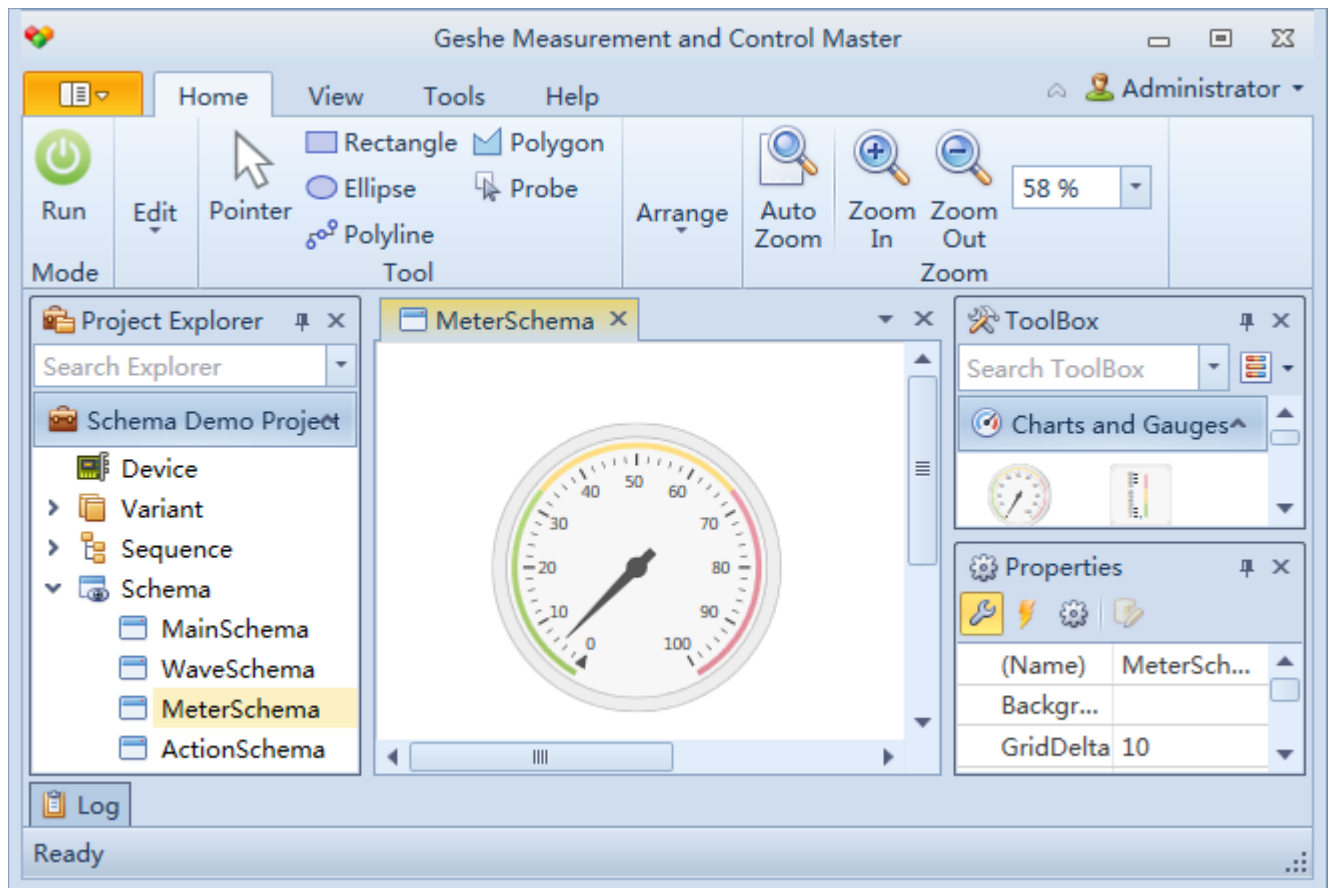


Command	Function
Calculator	Run the calculator of the windows system.
Checksum	Run Checksum Calculator.
CRC	Run CRC Calculator.
DES	Run DES Calculator.
Hash	Run Hash Calculator.
Base64	Run Base64 Calculator.
ASCII Checklist	Run ASCII Checklist.
Unicode Converter	Run Unicode Converter.

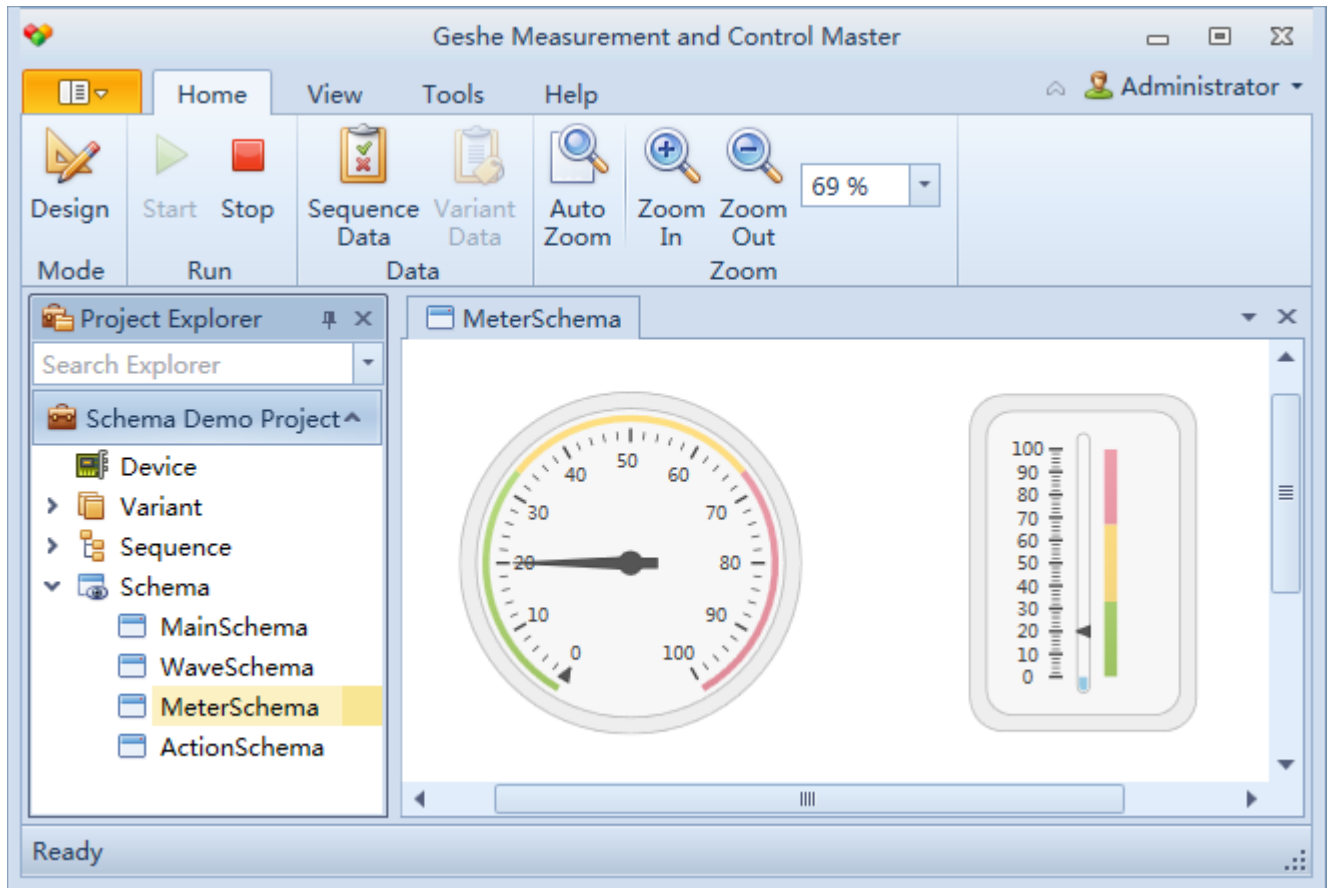
2.3.4 Editor Area

Editorial area can be used to edit project entries in the design state, such as sequence, variable, screen, etc. In the running state, it can be used to display project running data, running state and user interaction.

In the design state, when the item item item to be edited is opened, the corresponding editor is displayed in the editing area, the corresponding tool command button is displayed in the toolbar on the home page, the corresponding template is displayed in the toolbox, and the corresponding attribute set is displayed in the attribute view. As shown in the following figure, a screen is opened for editing. Many screen editing command buttons are added to the home toolbar. The toolbox loads the recently edited screen element template, and the attribute view loads the attribute set of the currently selected screen element.



In the running state, when the item item item to be displayed is opened, the corresponding page is displayed in the editing area, and the corresponding tool command button is displayed in the toolbar of the home page. Run a screen as shown in the following figure.



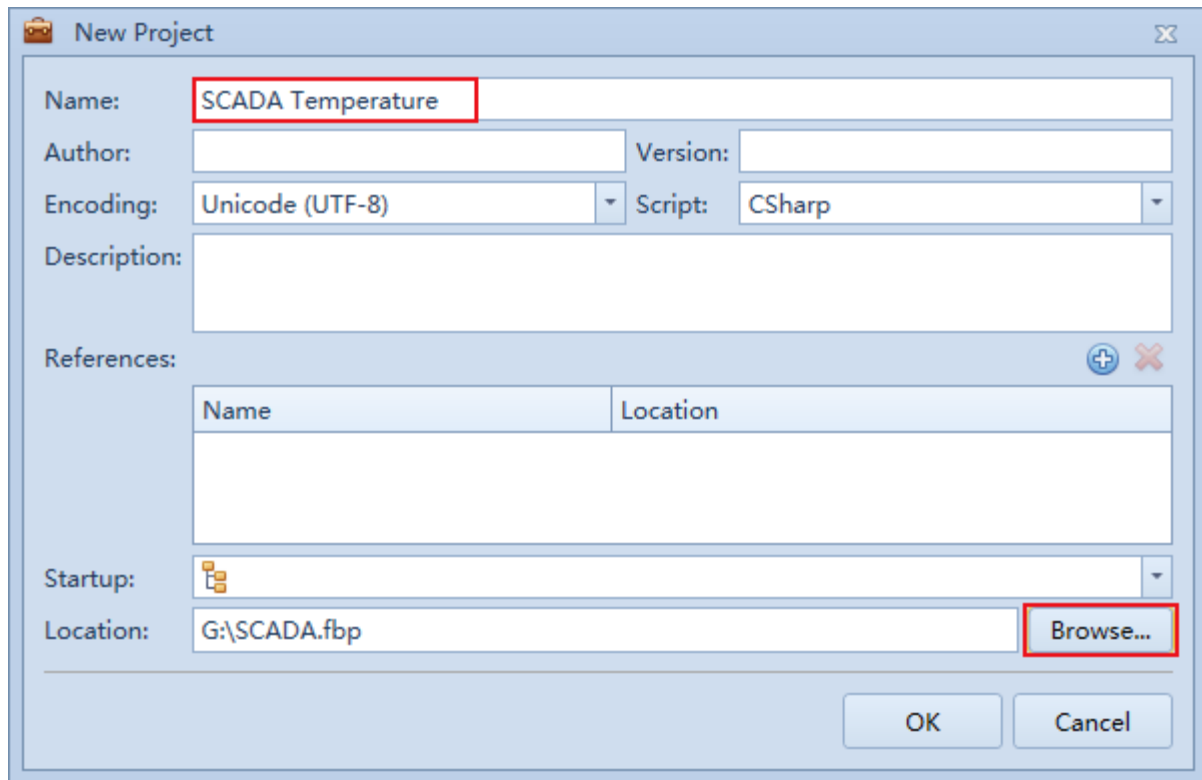
2.4 Create a Data Acquisition and Monitoring Project

A thermocouple acquisition module adopts Modbus RTU communication protocol, in which read temperature command is 03 command, address is 0000, temperature value is 16 bit signed integer.

This project demonstrates the reading temperature, then saves the acquisition time and temperature data into the text file variable "temperature data", and finally displays the temperature changing process with time by using the curve chart.

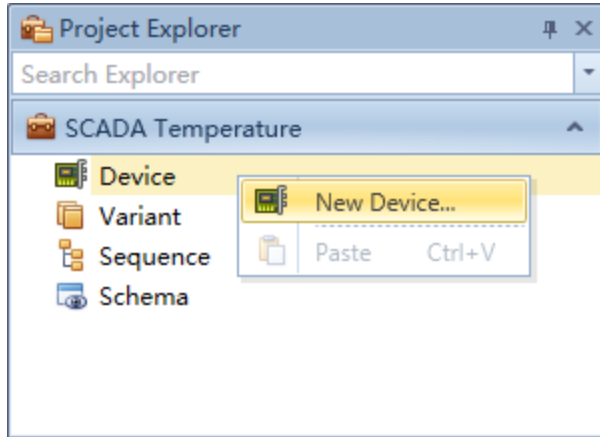
2.4.1 Step 1 – New Project

Start Geshe MC Master, select "New Project" from the menu in the upper left corner, then fill in the project name "Data Acquisition and Monitoring Demonstration Project" in the pop-up "New Project" dialog box, and then click "Browse..." Button, select the save path and fill in the project file name "Data Acquisition and Monitoring Demonstration Project", and finally click the "OK" button.

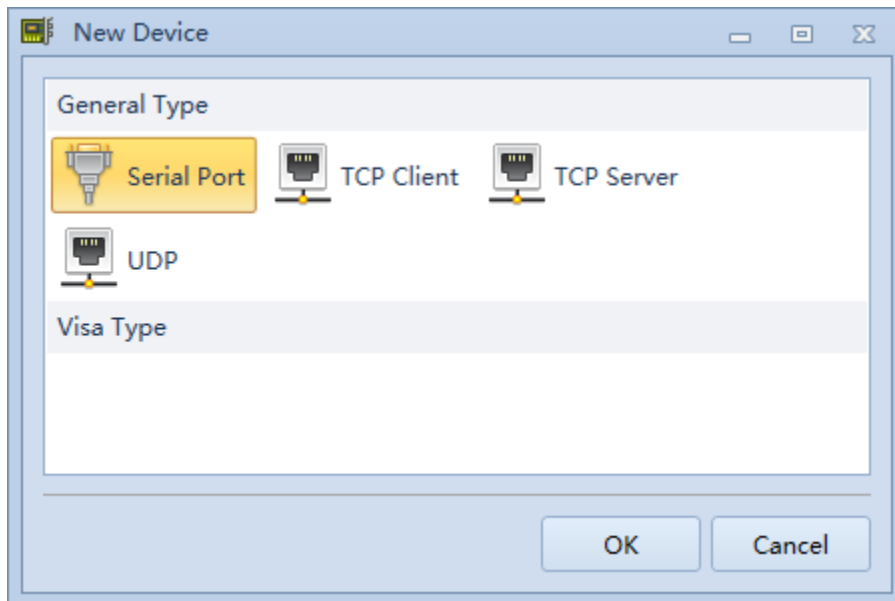


2.4.2 Step 2 – Add Serial Port Device

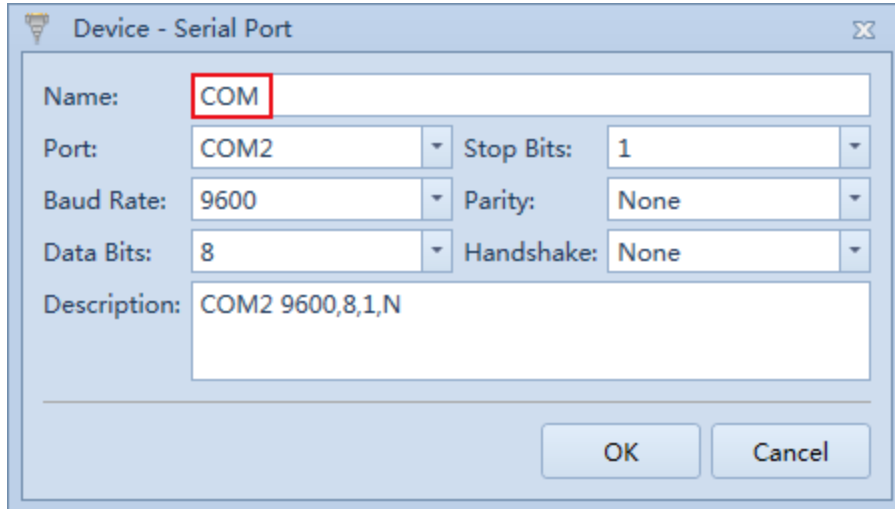
Select the "Device and Interface" node in the Project Manager, then click the right mouse button, and select "New Device..." in the pop-up menu.



In the new device dialog box, select "Serial Port" and click "OK".

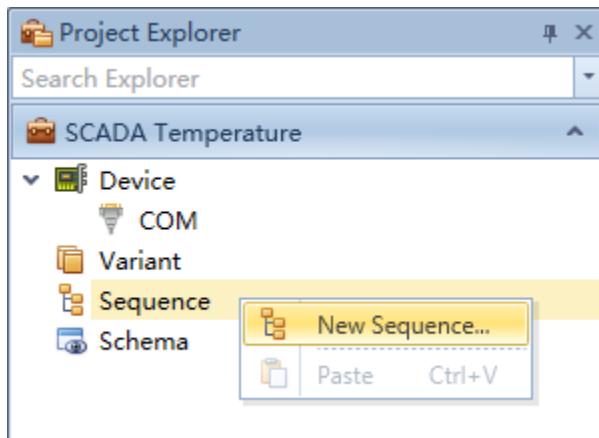


Pop up the device properties dialog box, fill in the "Nname" and other device parameters, and finally click the "OK" button. Among them, "device name" is the identification of the device, which can be any string. The device name must be used to refer to the device.

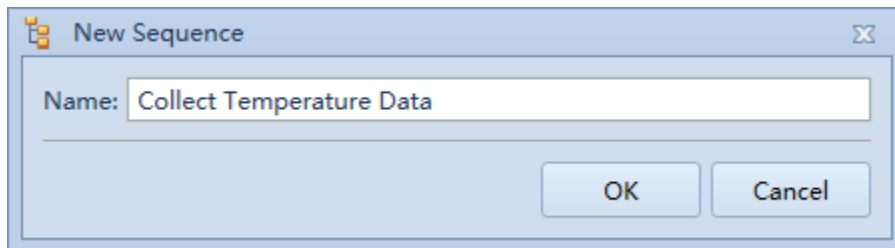


2.4.3 Step 3 – Add Sequence

Select the sequence node in the project explorer, click the right mouse button, and select New Sequence in the pop-up menu.



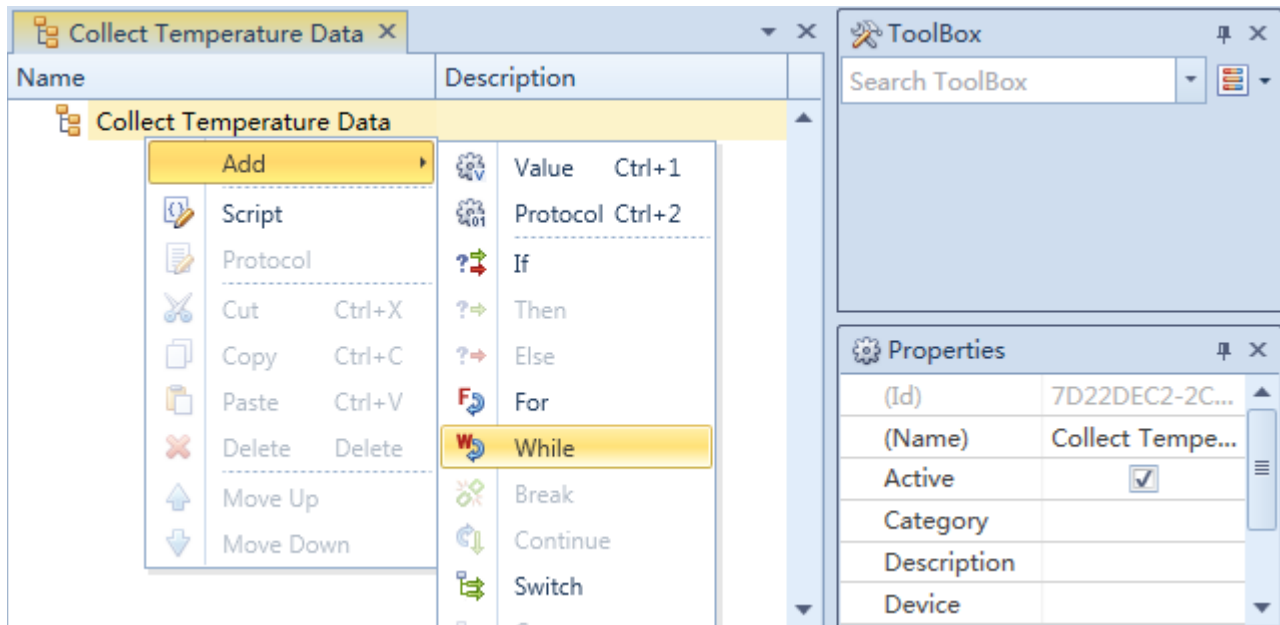
Pop up the new sequence dialog box, fill in the "Name" and click "OK".



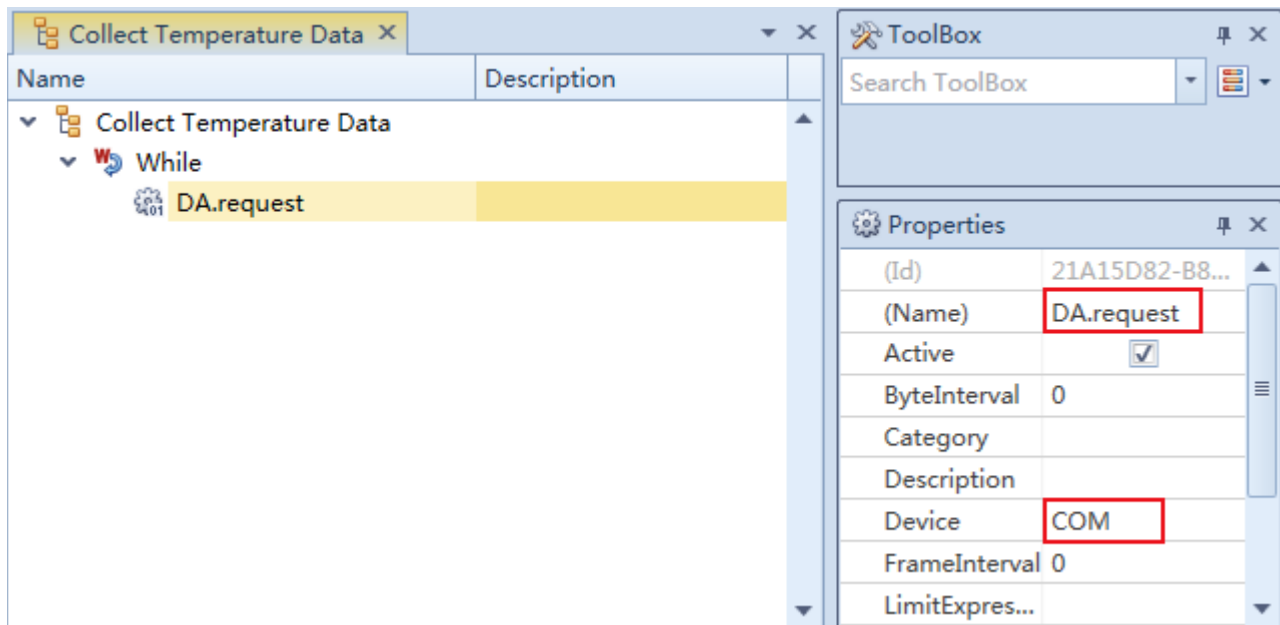
Select the newly created "Collect Temperature Data" node in the Project Explorer, then double-click the left mouse button, or right mouse button, and select "Edit" in the pop-up menu. Open the Sequence Editing page.

On the "Collect Temperature Data" editing page, select the "Collect Temperature

Data" node, click the right mouse button, select "Add - > While" in the pop-up menu, add the While type step, set the condition parameter ConditionExpression to True, and perform the acquisition task indefinitely.



Next, add the send command, select the "While" node, click the right mouse button, select "Add - > Protocol" in the pop-up menu, add the protocol type step, and then in the property panel, modify the step name "(Name)" to "DA.request", set the TransceiveMode to Send, indicating active sending, device property to select the device named "COM".



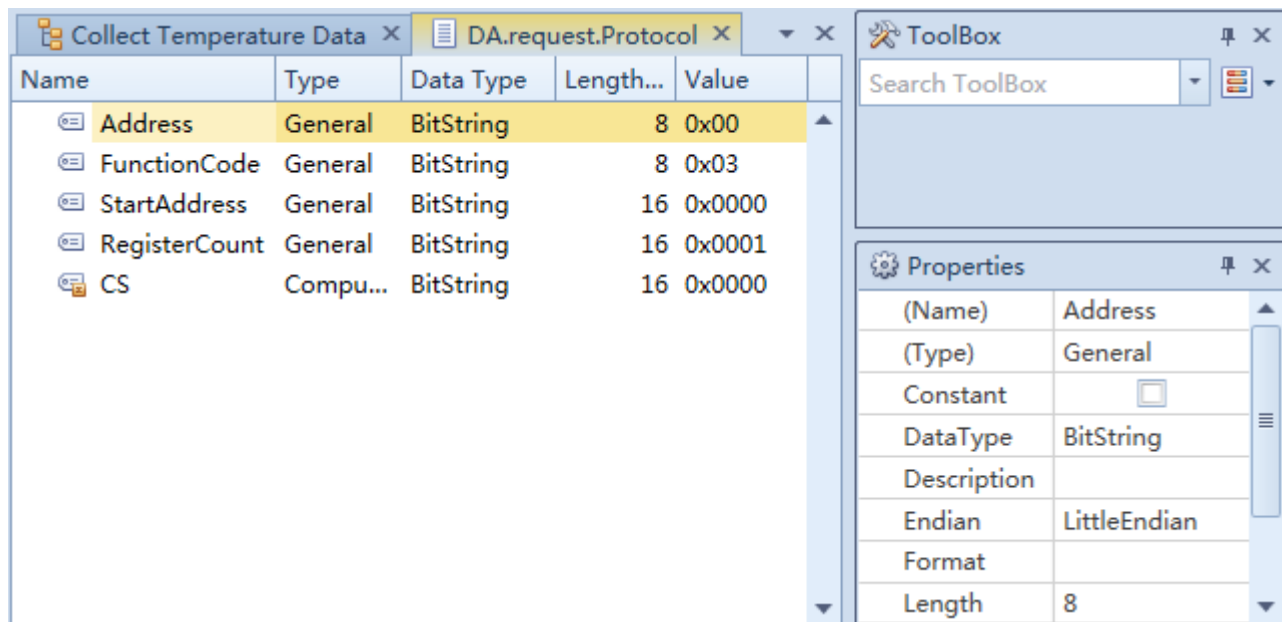
Next, edit the protocol data of the acquisition command, selects the "DA.request" node, click the "Protocol" button in the toolbar, open the protocol editing interface,

click the right mouse button, and chooses "Add - > Protocol Field" in the pop-up menu, and add protocol fields in turn by pressing the following property table.

(Name)	(Type)	Constant	DataType	Endian	Length	Value
Address	General		BitString	LittleEndian	8	0x00
FunctionCode	General	√	BitString	LittleEndian	8	0x03
StartAddress	General	√	BitString	BigEndian	16	0x0000
RegisterCount	General		BitString	BigEndian	16	0x0001
CS	Computable	√	BitString	LittleEndian	16	0x0000

Among them, the parameter configuration of the check code is as follows.

Property	Value	Description
Algorithm	CRC16MODBUS	Algorithm for Computing
Priority	1	Calculating priority, useful when there are multiple computational fields.
Location	Back	Represent a computational field after the data that needs to be computed.
StartPosition	0	Beginning calculation byte ordinal number
EndPosition	-1	End calculation byte ordinal number, -1 means calculation before the field.



Next, add the receive command, select the "While" node, click the right mouse button, select "Add - > Protocol" in the pop-up menu, and then in the property panel, modify the step name "(Name)" to "DA.response", and set the TransceiveMode to Receive, which means receive. Device property chooses the device named "COM".

Next, edit the protocol data that receives the data acquisition command, select the

"DA.response" node, click the "Protocol" button in the toolbar, open the protocol editing interface, click the right mouse button, select "Add -> Protocol Field" in the pop-up menu, and then add the protocol field according to the following property table.

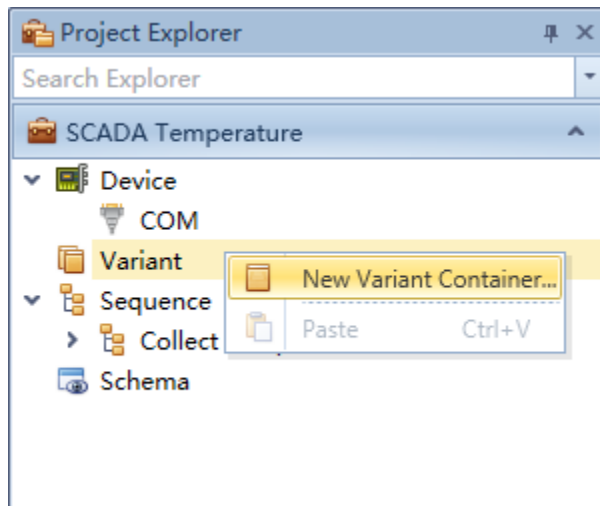
(Name)	(Type)	Constant	DataType	Endian	Length	Value
Address	General		BitString	LittleEndian	8	0x00
FunctionCode	General	√	BitString	LittleEndian	8	0x03
Bytes	General	√	BitString	LittleEndian	8	0x02
Temperature	General		Int16	BigEndian	16	0
CS	Computable	√	BitString	LittleEndian	16	0x0000

Among them, the parameter configuration of the check code is as follows.

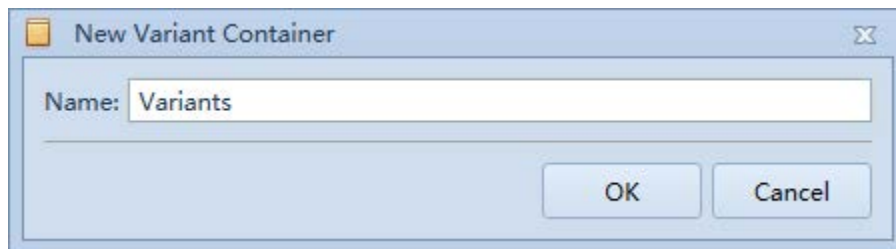
Algorithm	Priority	Location	StartPosition	EndPosition
CRC16MODBUS	1	Back	0	-1

2.4.4 Step 4 – Add Variants

Select the "Variant" node in the Project Explorer, then right-click the mouse, and select "New Variant Container..." in the pop-up menu.

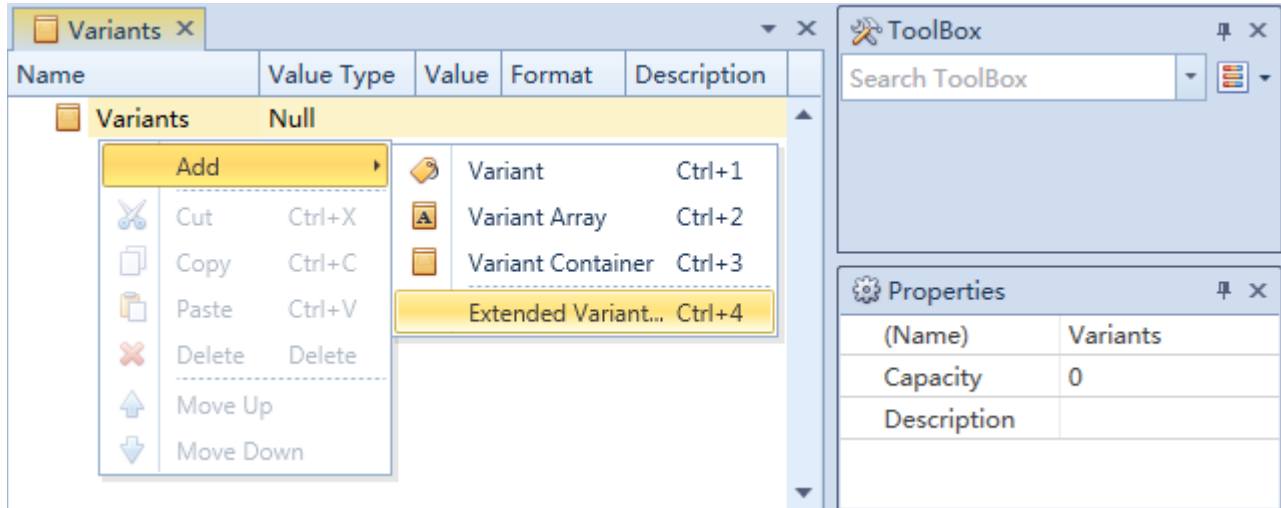


Pop up the new variant container dialog box, fill in the "Name" and click "OK".

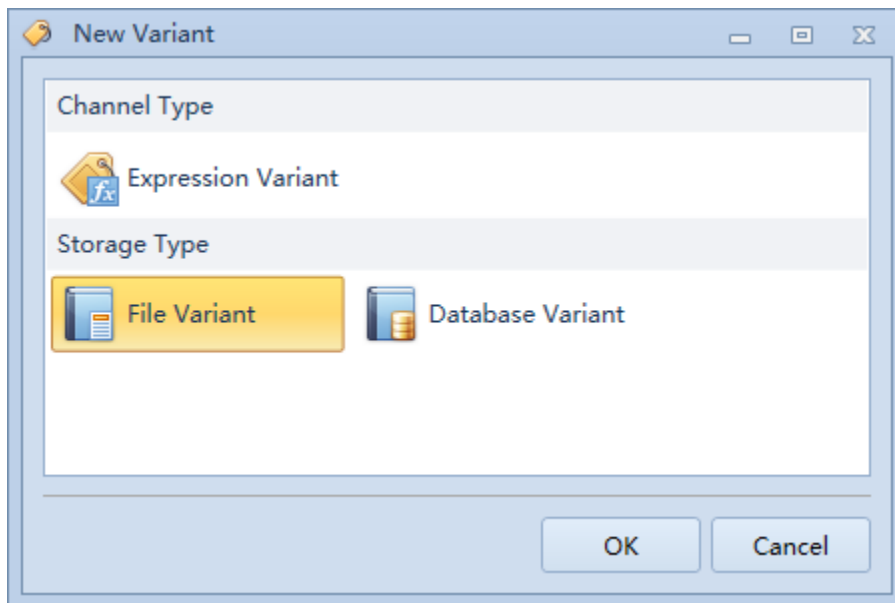


Select the newly created "Variants" node in the Project Explorer, then double-click the left mouse button, or right mouse button, and select "Edit" in the pop-up menu. Open the variant editing page.

On the editing page of "Variants", select the node of "Variants", right-click the mouse, and select "Add -> Extended Variant..." in the pop-up menu.



In the New Variant dialog box, select File Variant and click OK.



Next, select the newly created text file variant, and in the property panel, modify the property values according to the following property table in turn.

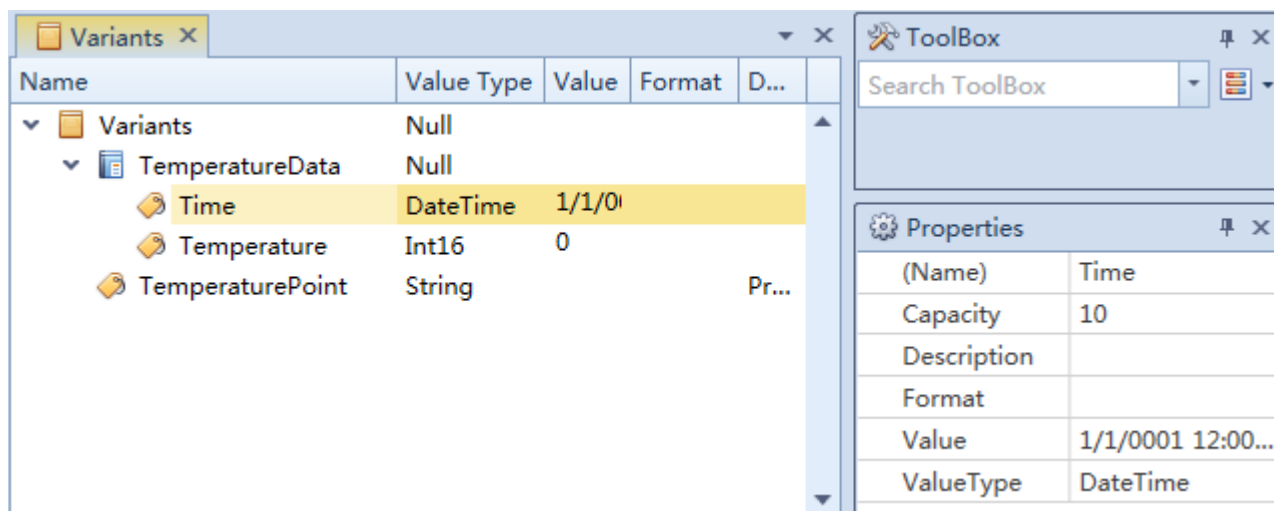
Property	Value	Description
(Name)	TemperatureData	Variant name
Capacity	10	Variant cache capacity, the higher the acquisition speed, the larger the cache.

Directory	D:\Temp	Directory of file storage
FileName	TemperatureData.txt	Filename

Next, select the "TemperatureData" node, click the right mouse button, select "Add -> Variant" in the pop-up menu, add two variables in succession, named "Time" and "Temperature", in the attribute panel, modify the attribute values according to the following attribute table in turn.

Property	Value	Description
(Name)	Time	
Capacity	10	
ValueType	DateTime	

Property	Value	Description
(Name)	Temperature	
Capacity	10	
ValueType	Int16	



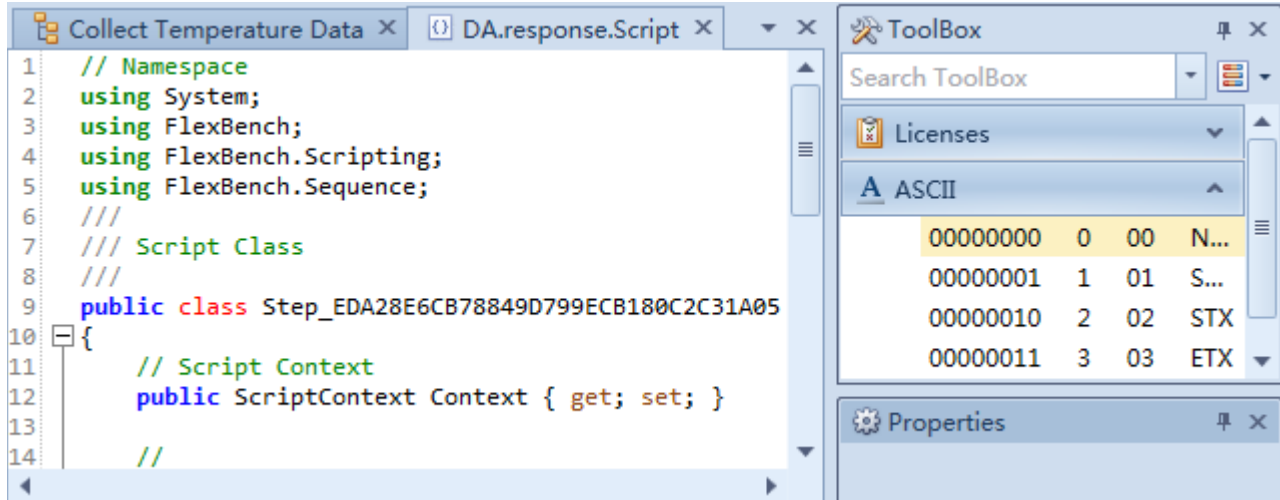
Text file type variant, when all its sub-variables have changed, forms a record and saves it to the file. The temperature data in this example, after each acquisition, can generate a record by assigning the acquisition time and temperature value to "Time" and "Temperature" variants respectively through script.

2.4.5 Step 5 - Use Scripts to Associate Sequence Data with Variables

The temperature data collected in the sequence can be assigned to the variants in the variant table by script, and then the collected data and variable values can be correlated. The script types supported by the software system are C#, Visual Basic and Python, which can be set in the project properties. This example is demonstrated with C# script.

On the "Collect Temperature Data" sequence editing page, select the "DA.response"

node and click the "Script" button in the toolbar to open the script editing page.



In the open script editing page, the script code of the "DA.response" step is displayed. Below is the step script template for C #.

```
// Namespace
using System;
using FlexBench;
using FlexBench.Scripting;
///
/// Script Class
///
public class Step_EDA28E6CB78849D799ECB180C2C31A05
{
    // Script Context
    public ScriptContext Context { get; set; }

    // Execute before execution of the step
    // Parameters: context – Step runtime context
    //              step – current executing step
    // Return: Temporarily undefined
    public Int32 BeginExecute(FlexBench.Sequence.IStepContext context,
        FlexBench.Sequence.IStep step)
    {
        return 0;
    }

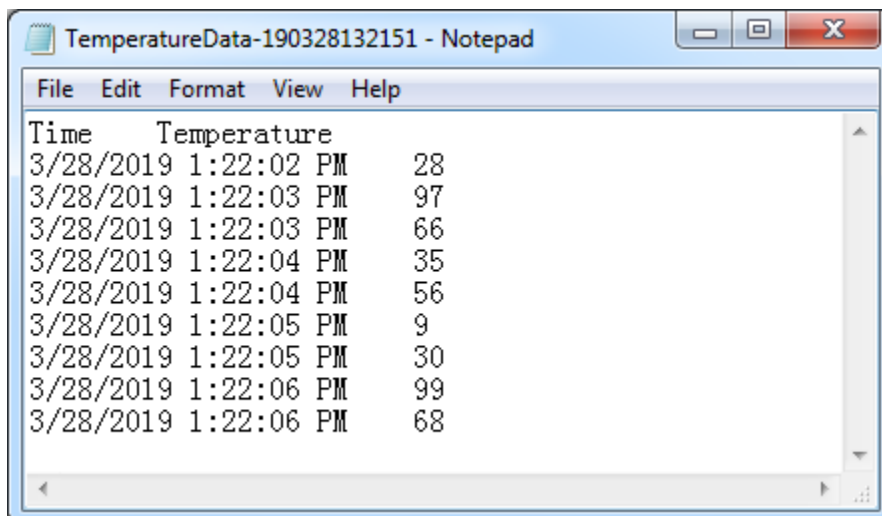
    // Execute after execution of the step
    // Parameters: context –Step runtime context
    //              step –current executing step
    // Return: Temporarily undefined
    public Int32 EndExecute(FlexBench.Sequence.IStepContext context,
```

```
FlexBench.Sequence.IStep step)
{
    return 0;
}
```

Next, in the EndExecute function, the collected data is extracted and assigned to the specified variants.

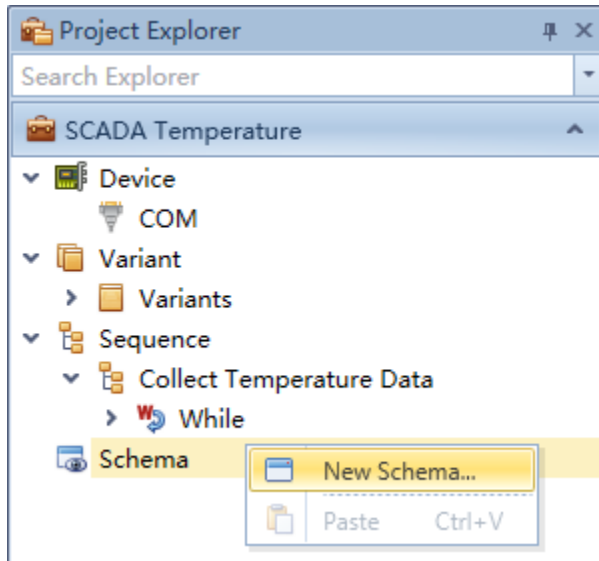
```
public Int32 EndExecute(FlexBench.Sequence.IStepContext context,
    FlexBench.Sequence.IStep step)
{
    if (step.Result.Status == (int) FlexBench.Sequence.ResultStatus.Passed)
    {
        // Extract the collected temperature values, where the Visible = True values of
        // all protocol fields are stored in the Data Fields container.
        Int16 temperature = (Int16)step.Result.DataFields[0].Value;
        DateTime time = DateTime.Now;
        // Set the variable value of the variable table, which is a Variants container,
        // accessed by path.
        context.Variants["Variants/TemperatureData/Time"] = time;
        context.Variants["Variants/TemperatureData/Temperature"] = temperature;
    }
    return 0;
}
```

So far, the function of collecting and saving temperature data has been completed, and the data format saved is shown in the following figure.

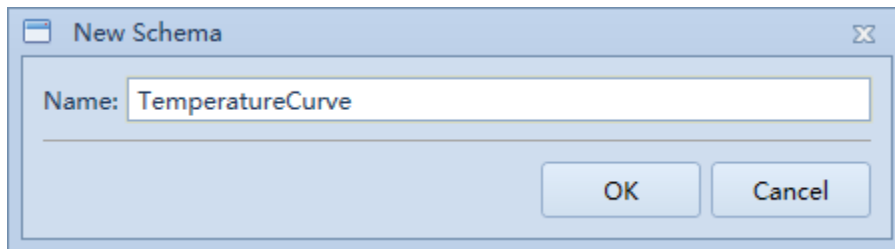


2.4.6 Step 6 – Add Schema

Select the "Schema" node in the Project Explorer, then click the right mouse button, and select "New Schema..." in the pop-up menu.

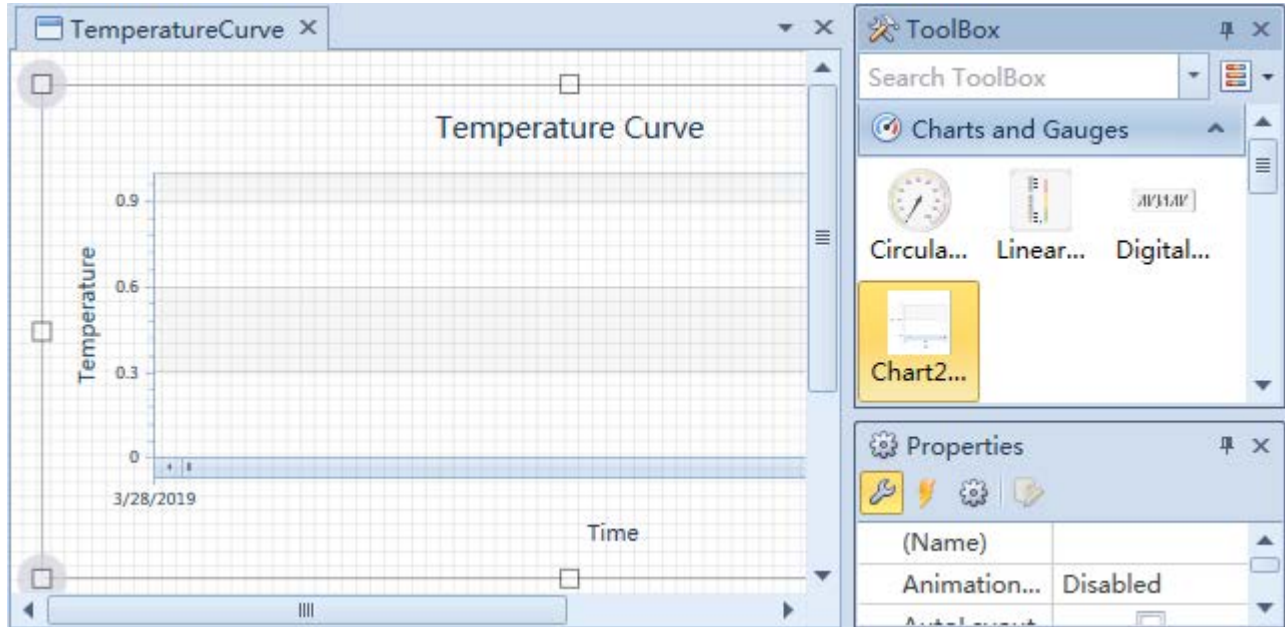


Pop up the new schema dialog box, fill in the "Name" and click "OK".



Select the newly created "TemperatureCurve" node in the Project Explorer, then double-click the left mouse button, or right mouse button, and select "Edit" in the pop-up menu. Open the schema editing page.

Click on the menu in the upper right corner of the toolbox and select "Open Mould File..." from the pop-up menu. In the Controls directory, select the template file "ChartsAndGauges.fbschema" and click "Open"; then use the mouse to select the "Chart2DLine" entry in the toolbox and create a curve display control in the schema.



Next, in the property panel, modify the property values of the control according to the following property tables in turn.

Property	Value	Description
AxisXScaleType	DateTime	X-axis data type
AxisXTitle	Time	X-axis title
AxisYTitle	Temperature	Y-axis title
Title	Temperature Curve	Title
DataEntry1		Data source 1, By binding data methods and associating variables.

Next, we deal with the problem of data binding. The type of data source of the curve control is a string in the format of “<serie name>,<X value>,<Y value>”, where X value is date format or digital format, and Y value is digital format. For example, to display the point of a curve named “Temperature Curve” (2019/1/1 10:10:10,123), the value of the data source is “Temperature Curve,2019/1/1 10:10:10,123”.

Therefore, to create a new string type variant to provide data for the curve display control, in the “Variants” edit page, select the “Variants” node, click the right mouse button, select “Add -> Variant” in the pop-up menu, named “TemperaturePoint”, in the property panel, in turn, according to the following property table to modify the property value.

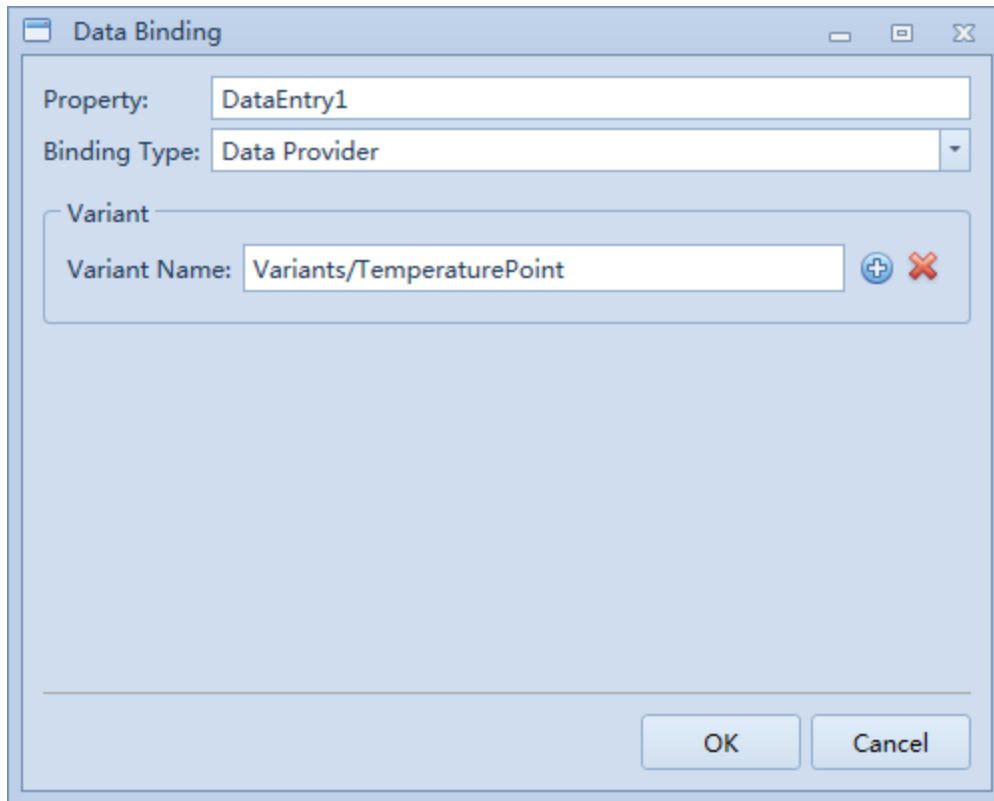
Property	Value	Description
(Name)	TemperaturePoint	
Capacity	10	
ValueType	String	

Next, the “TemperaturePoint” variant is assigned in the EndExecute function of the script code of the “DA.response” step.

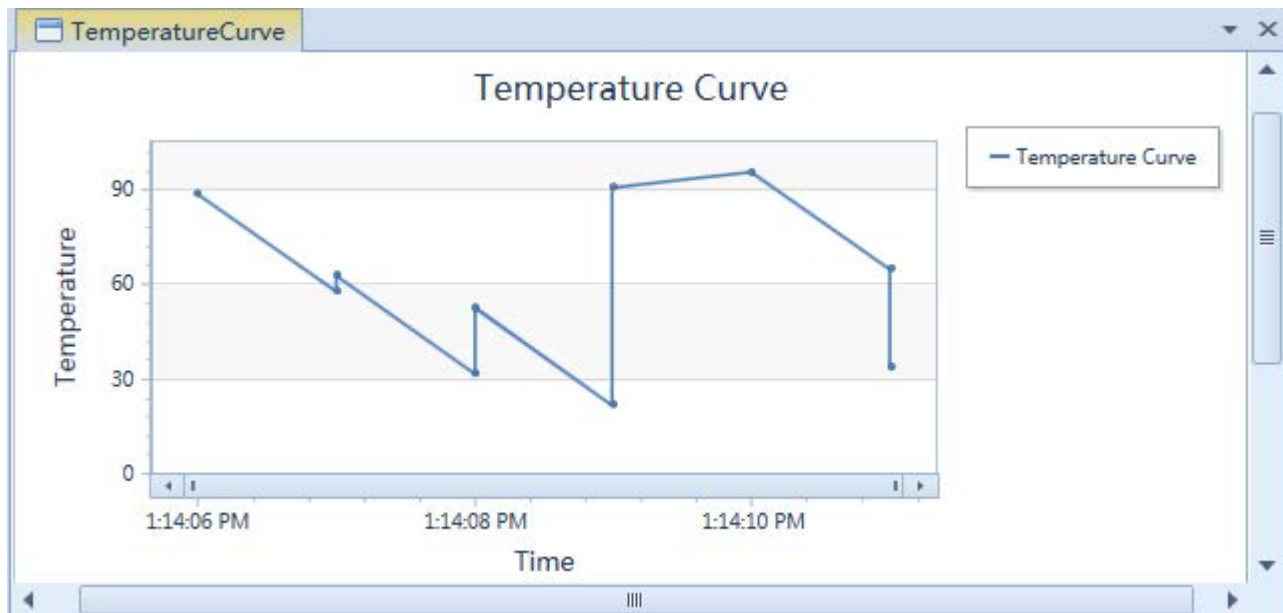
```
public Int32 EndExecute(FlexBench.Sequence.IStepContext context,
    FlexBench.Sequence.IStep step)
{
    if (step.Result.Status == (int) FlexBench.Sequence.ResultStatus.Passed)
    {
        // Extract the collected temperature values, where the Visible = True values of
        // all protocol fields are stored in the Data Fields container.
        Int16 temperature = (Int16)step.Result.DataFields[0].Value;
        DateTime time = DateTime.Now;
        // Set the variable value of the variable table, which is a Variants container,
        // accessed by path.
        context.Variants["Variants/TemperatureData/Time"] = time;
        context.Variants["Variants/TemperatureData/Temperature"] = temperature;

        // Set TemperaturePoint variable
        string temperaturePoint = "Temperature Curve," + time.ToString() + "," +
        temperature.ToString();
        context.Variants["Variants/TemperaturePoint"] = temperaturePoint;
    }
    return 0;
}
```

Next, select the control in the "TemperatureCurve" schema editor, select "DataEntry1" in the property panel, click the "Create Data Binding" button in the property toolbar, pop up the data binding dialog box, select the "Data Provider" for the binding type, then click the "Add Variant" button, and select the "TemperaturePoint" variant in the pop-up variant selection dialog box. Click OK.



So far, the temperature curve is completed.



3. FAQ