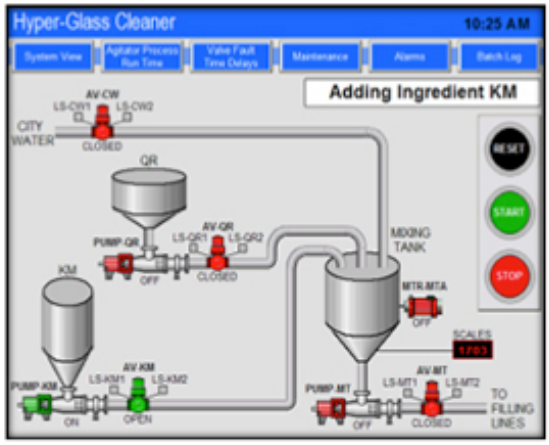
How To Program RSView32



How to Program an HMI and SCADA System with Rockwell Automation's RSView32

By Neal Babcock

How to Program RSView32

How to Program an HMI and SCADA System with Rockwell Automation's RSView32

By Neal Babcock

Industrial Automation Series

engineer-and-technician.com

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Contents

Introduction....1
PLCs....2
Hardware....3
SLC Rack....3
SLC Power Supply....3
SLC Processors....4
SLC I/O Modules....4
Ladder Logic....5
The Dialect of PLCs....5
Project Scope....8
Summarizing the Scope....17
Beginning the Project....18
Tags and the Tag Database....23
Digital Tags....24
Timers....28
Analog Tags....30
Creating the Screens....34
Screen 1 – System View....36
Designing the Master Layout....38
Color Blindness....39
Designing the Header....40
The Navigation Menu....45
Screen 1 Content....69
Pump Icons....77
Agitator Motor....80
Scales....81
Adding Piping....83
Review of the System View Display....86
Configuring the Menu....88
Screen 2 - Agitator Process Run Time Display....92
Screen 3 - Valve Fault Time Delays....96
Screen 4 - Maintenance Display....98
Screen 5 - Alarms Display....101
Alarm Setup....106
Configuring a Tag to Trigger an Alarm....107
Screen 6 - Batch Log display....110
The Final Result....115
Finishing Touches....119
Trending....123
Tips and Tricks....125
The Steps to a Successful and Profitable RSView Project....128

Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Introduction

The purpose of this book is to teach you how to design and program an HMI or a SCADA system with RSView32.

There is a sample project included that contains a Project Scope. The Project Scope (or Functional Specification, or whatever your company might call it) defines in detail how the system is to operate when the project is finished.

You will learn how to take a Project Scope and turn it into a working RSView program.

It will show you the keystrokes and mouse movements that you need to know to use RSView.

Finally, it provides a number of tips that will save you a bunch of time.

This book assumes you have a little background with PLCs – perhaps you have worked with other PLCs from other manufacturers or you have helped to install and wire PLCs. Perhaps you are a Mechanical, Chemical or Process Engineer and you need to learn how to use RSView32.

If you need a more thorough understanding of basic PLC concepts, you might want to consider the Beginner's Guide to PLC Programming *How to Program a PLC (Programmable Logic Controller).* This ebook, along with the online tutorial, provides an example of how to automate a drill press, while explaining all the basic concepts of PLC programming that are necessary to write a solid PLC program.

Visit engineer-and-technician.com/ if you would like to learn more about this book.

For now, it should be enough to say that the PLC receives signals and information from hardwired external devices, such as the position of a limit switch, the speed of a motor or the numeric value of a weight sensor.

This information is linked from the PLC to RSView through *tags*. Tags are simply memory locations to which both the PLC and RSView have access. Some tags are created in RSView; most are created in the PLC.

Understanding tags and the manipulation of tags is a fundamental skill required for HMI or SCADA programming.

We will also discuss the aesthetic appeal, or "looks" of HMI designs. This attribute is often underrated. For the time being, just keep in mind that an HMI that looks good, while still providing the functionality required, is easier for an operator to use. Because it is visually

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

appealing, screens and navigational paths will be easier to remember. We'll cover more of this later.

Rockwell Automation Technical Support

Unfortunately, we can't anticipate all the problems you might face as you are troubleshooting a program on the factory floor. There are just too many variables. This is why you must establish a relationship with your local Rockwell Automation technical support team. Get to know them *before* you are in the final stages of a start-up and you run into a problem. They are very helpful and they can save you hours of frustration.

Many Rockwell reps are not just technical support personnel; they are skilled engineers that are responsible for running their own projects and writing and troubleshooting their own programs. If you run into a problem, more than likely they have already seen it and have come up with a solution.

PLCs

Nearly all the industrial equipment that you find in a modern manufacturing facility shares one thing in common - computer control. The most commonly used controller is the PLC, or the Programmable Logic Controller, using a programming language called Ladder Logic. The language was developed to make programming easy for people who already understood how switches, relay contacts and coils work. Its format is similar to the electrical style of drawing known as the "ladder diagram".

The most popular and most widely used manufacturer of PLCs is Rockwell Automation, who produces the Allen-Bradley SLC series of PLCs. The MicroLogix and SLC families of processors and I/O modules are all programmed using Rockwell's proprietary software known as RSLogix (RSLogix 500 is the software used to program the SLC family of PLCs, while RSLogix 5000 is used for the ControlLogix PLCs).

The SLCs are an older generation of Allen-Bradley PLCs. They are being slowly replaced by the ControlLogix and CompactLogix lines. However, the SLCs are still very popular, as they are powerful PLCs that still have many applications. In addition, there are many plants that use SLCs that don't want to upgrade to ControlLogix processors, because of training issues and the higher initial cost.

Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Hardware

One of the nice things about Allen-Bradley's smaller PLCs is the relative simplicity of assembling the hardware to create a system.

First, let's see what it takes to assemble an SLC 500 system. You only need to have a few components: a rack, a power supply, a processor and some I/O modules.

SLC Rack

These come in four configurations, with varying capacities for installing the I/O modules.

1746-A4	4-Slot chassis
1746-A7	7-Slot chassis
1746-A10	10-Slot chassis
1746-A13	13-Slot chassis

A ***rack*** is a frame that holds the modules of an SLC 500 system. It is similar to the motherboard and case in your personal computer. It provides a physical structure to hold the modules that create a system, like your computer's case. It also provides an electronic back plane that allows modules to communicate and interact.

In an SLC system, the SLC 500 processor always resides in Slot 0, which is the first slot.

SLC Power Supply

Power supplies come in varying capacities.

1746-P1
1746-P2
1746-P3
1746-P4
1746-P5
1746-P7

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Ladder Logic

Before we open RSLogix 500 and start programming, there are a few things you need to know about PLCs in general. I have summarized the basic terms and techniques required to work with ladder logic. It isn't a comprehensive summary, but if you are just starting out, the information here book will be very helpful.

Every PLC programmer, no matter what skill level, must know the principles described in this section and the Equivalent Logic section. There is simply no way around it.

To effectively write a program, or even edit one, the programmer must know how to visualize the effects of the changes he will make.

In other words, you have to be able to look at the logic "on paper" and imagine how the logic will work when it is entered into the PLC.

The Dialect of PLCs

Lets' define some terms and symbols:

INSTRUCTION – RSLogix's command language is comprised of "instructions". An XIC (it looks like a normally open contact **--] [--**) is an instruction. A timer is an instruction. A few of the most common instructions are described below.

BIT - an address within the PLC. It can be an input, output or internal coil, among others.

In RSLogix 500, there are a couple of ways to show the address of a bit. The default is:

[type]:[word]/[bit]

For example, an address that references an output of an SLC 500 is O:5/0. That is:

O:5/0 means that it is a physical output.
O:**5**/0 means that it uses Slot 5 (the 6th physical slot) in the rack.
O:5/**0** means that it is the first output on the card.

Remember that the first slot in an SLC 500 rack is Slot 0. That means a card that is installed in the 6th physical slot is addressed as Slot 5.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

An XIC instruction can reference a hardwired input, a hardwired output, an internal coil or a timer done bit, among others.

--]/[-- **Normally Closed Contact**
This is an inverted normally open contact.

When used with a hardwired input, this instruction is "true" until there is a voltage applied to the input. It then goes low, or off, and becomes “false.”

It also can be used with an internal coil, becoming true when the coil is off and becoming false when the coil is on.

Allen-Bradley calls these normally closed contacts “XIO”, or “e**X**amine **I**f **O**pen” instructions.

-( )- **Output Coil**
When used with a hardwired output, this function is off until the logic in the program allows it to turn on. It then becomes “true”, and will energize the device that is wired to the respective output.

If it is used as an internal coil, it will toggle the instructions associated with it. That is, it will close a normally open instruction and open a normally closed instruction.

Allen-Bradley calls these outputs “OTE”, or “**O**utpu**T** **E**nergize”.

An OTE may be used with a hardwired output or an internal coil.

TRUE - A state that indicates an instruction is allowing logic to “flow” through it.

Also, if the logic in a rung turns on the output of the rung, then the rung is said to be true.

FALSE - Without stating the obvious, this is the opposite of true.

OK, that was a lot to cover and for you to understand – don’t worry, this will start getting easier.

Project Scope

We will use a batching operation as an example. Batching, as you may know, is the term that describes the mixing of assorted ingredients to make a finished product.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

7.4.4. LS-CW1 will verify that the valve is closed within a specified time after the valve was told to close. If the valve closure is not verified within the specified time, a fault will be generated, the system will shut down and the HMI displays the text “ALARM”.

7.4.5. The time delays used in the Valve Fault detection logic are individually adjustable in the HMI from 1 to 10 seconds.

7.4.6. NOTE: All valves and their respective limit switches work in the manner described above.

7.4.7. After the City Water has been added, valve AV-CW closes. The HMI no longer displays the text “ADDING WATER”.

7.5. Step 2 – Ingredient QR

7.5.1. Valve AV-QR is opened. After the valve position has been verified by LS-QR2, PUMP-QR pumps 390 lbs. of ingredient QR into the Mixing Tank. The HMI displays the text “ADDING QR”.

7.5.2. After the ingredient QR has been added to the Mixing Tank, PUMP-QR stops, valve AV-QR closes and the HMI no longer displays the text “ADDING QR”.

7.6. Step 3 – Ingredient KM

7.6.1. Valve AV-KM is opened. After the valve position is verified by LS-KM2, PUMP-KM pumps 173 lbs. of ingredient KM into the mixing tank. The HMI displays the text “ADDING KM”. This text is displayed while the pump is running.

7.6.2. After the ingredient KM has been added to the Mixing Tank, valve AV-KM closes. PUMP-KM stops. The HMI no longer displays the text “ADDING KM”.

7.7. Step 4 – Mixing

7.7.1. After LS-KM1 indicates the valve has been closed, the agitator motor MTR-MTA starts. The HMI displays the text “BLENDING”.

7.7.2. The agitator runs for 3 minutes.

7.7.3. After the agitator is finished, the HMI no longer displays the text “BLENDING”.

7.8. Step 5 – Pump to filling lines

7.8.1. Valve AV-MT will open. PUMP-MT turns on and pumps the entire batch to the filling lines. The HMI displays the text “PUMPING TO LINES”.

Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

END OF HYPER-GLASS CLEANER BATCHING PROJECT SCOPE

Summarizing the Scope

So, what did we get from the scope? Let's summarize:

System Control
From an operational and programming standpoint, 1275 lbs. of water will be added to the Mixing Tank. Then, 390 lbs. of QR will be added. The last ingredient is KM, of which we will add 173 lbs.

After all the ingredients are in the Mixing Tank, we have to blend it.

After the batch is blended, we will the pump the finished product in the tank to the filling lines.

We have to make sure all the valves open or close as commanded. If they do not, then we need to shut down the process.

We need to make sure the level in the Mixing Tank doesn't get too high. If it does, we must shut down everything.

We need to make sure that the respective valves for the pumps are open before we turn on the pumps.

Though you may not do the PLC programming, it behooves you to understand the process.

System Monitoring
From an HMI and SCADA standpoint, we see there are 6 screens required. The screen resolution is 1024 x 768.

The monitor is a touchscreen. It would be best to have this monitor available to you during development.

All colors are defined in term of RGB values. This is very helpful, and will save a lot of headaches further on in the project, as there will be no conflicts over colors as long as we stick with the RGB values we area given.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

The icons that we will use on the screens are defined.

We will need to be able to change a few registers in the PLC through the HMI for the agitator run time, valve fault times, etc.

Make sure that you clearly understand the scope of any project that is given. If there are questions or gray areas, present these issues to your project manager or your client before you begin development.

Beginning the Project

Before you start working on the HMI, you will need an electronic copy or a printed copy of the PLC program, including a list of all of the data files and the list of tags.

The first thing we must do is to create the project.

It is also best if you are connected to the PLC already. The screens below assume that. However, if you are not connected, you can skip some of these steps until later.

Begin by opening RSView32.

Click Start > All Programs > Rockwell Software > RSView32 > RSView32 Works

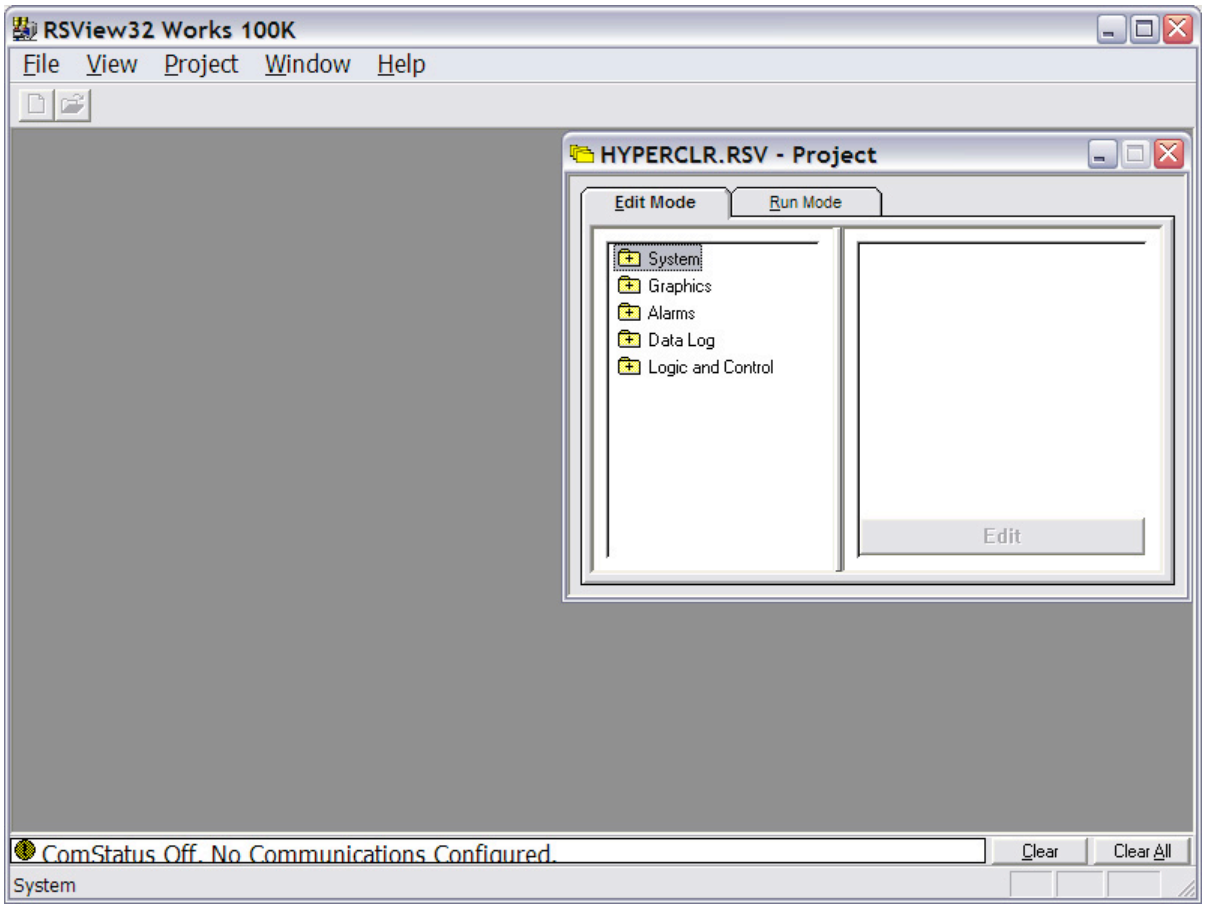
After RSView32 opens, click File > New

A dialog box will open. Make sure you choose your folder well; sometimes, you will run into problems if you decide to mode the folder after you have started a project. There are relative paths that RSView defines, so plan ahead. To be safe, you might want to create a folder in your root directory that is dedicated to your project.

In this case, we will call our project “hyperclr”. Click on the “New Folder” icon and fill in the fields. Click “Open”.

Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

This screen will appear.



Stretch the Project window down so that you may see more of the folders the window.

Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

These are all the basic settings you need to begin screen development. We will go over the other functions in the Project tree later on.

Tags and the Tag Database

This is a very important section of this book. To fully understand how RSView functions with a PLC, you must understand how tags work and how to manage and edit the tag database.

There is a list of the tags used in this project in Excel format included with this book called hyperclr-Tags.xls. Take a moment to look over this spreadsheet and understand the attributes of each tag.

Tags are the links that connect RSView with the PLC; the tags are common to both the PLC and RSView. The PLC and RSView have access to the values held by the tags.

The value of the tag in RSView reads or writes the value of the bit (or the word) in the PLC, and vice-versa.

When you define a tag that references a bit in the PLC, every time the bit in the PLC turns *on* the tag value in RSView will change from 0 to 1.

Conversely, every time the bit in the PLC turns *off*, the tag value in RSView will change from 1 to 0.

If you define an analog tag, such as the weight of the mixing tank in our example, as the value of the word in the PLC changes, the value of the associated tag in RSView will change.

You can also use RSView to send values to the PLC.

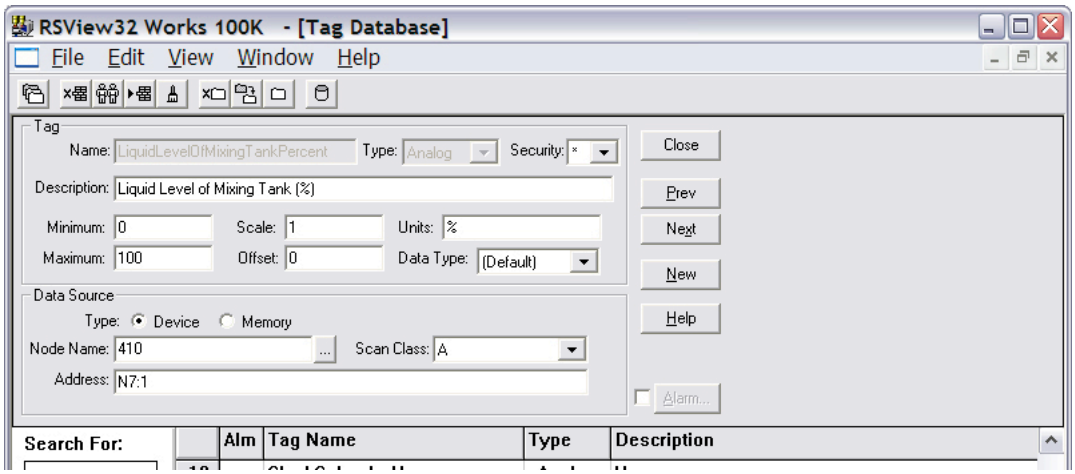
However, if the ladder logic is such that the value of the bit or word is determined by the ladder logic, the value you send to the PLC will be overwritten. The PLC has priority. We will cover this aspect later in the book.

Let's see how to create a tag and tie it to a bit in the PLC.

Look at the RSLogix printout for the Batching project. Find the section called "RSLogix 500 Cross Reference Report - Sorted by Address". This section shows all the outputs, inputs, internal bits, timers and words that we want to use to create tags in RSView.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

The address is N7:1. It will look like this when you are done.



You have completed a major phase of this project. Entering the tags is a tedious, but necessary, part of the process. It starts to get a little more interesting from this point on.

Creating the Screens

Before you start creating screens in RSView, you will need to have a general idea of what will be on each screen. Spend a few minutes sketching out the screens on a full size note pad – this will help you visualize the entire project and save time as you generate the screens in RSView.

As you develop the screens, it is much faster if you set your monitor resolution higher than the resolution required by the project. This lets you see the whole window of the screen you are developing, without having to constantly use the scroll bars.

Let's review the HMI Screens section of the scope to find out how many screens we will need. I have created a table the shows the scope requirements on the left and the required RSView screen on the right. This is another way of dissecting the scope to make sure we catch everything.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Even if a header is misplaced by one pixel, it will appear to “jump” when you are going from screen to screen. This is undesirable and certainly not very professional.

The overall size of the header is important, usually with smaller being better. Don’t let it overwhelm a screen. Besides, a large header will make things difficult as you try to add content (valve icons, pump icons, etc.) to your screen. You will run out of room.

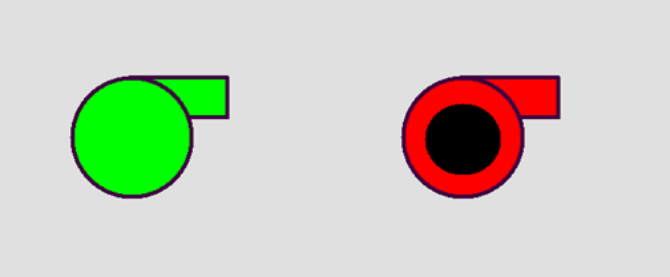
It may be useful to talk with a graphic designer during this process. They have studied graphic layouts and designs, and may offer some good advice.

However, on the other hand, realize that sometimes graphics designers focus too much on looks and sacrifice functionality. Dark gray text on a light gray background may be artistically appealing, but it is much more important for the operator to be able to read the display from 10 feet away. Use your good judgment.

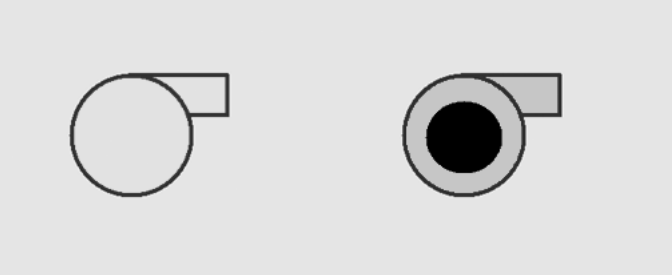
Color Blindness

Before we get too far into our master layout, let’s think about color blindness, as it is prevalent among many men. Though our project scope did not address this, as designers we must take this into consideration.

There are a couple of ways to get solve this problem. We could use icons that have different shapes, depending on whether they are on or off. In addition to being a different color, a pump icon that is off could have a round hole cut out in the middle, such as is shown below.



Here is what the pumps shown above might look like to a person who is colorblind.



How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

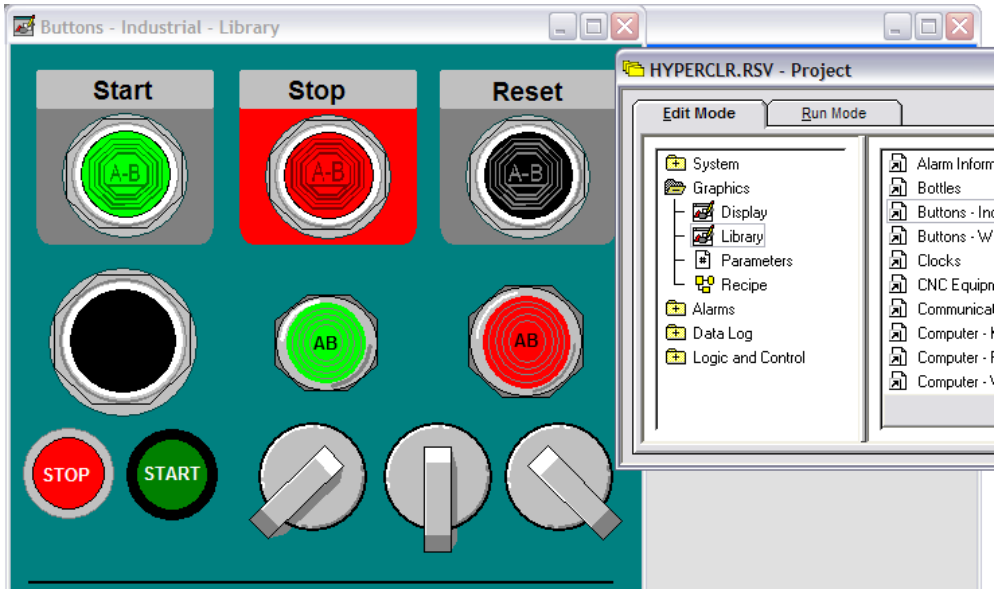
Resetting the Alarm

Since the “SystemFault” bit is latched on in the PLC when there is an alarm, we need a way to clear the alarm. In looking at the PLC program, we see that the “ResetAlarm” bit will accomplish that.

We can make an object to do just that. In an effort to use objects from the standard RSView library, we will use a pushbutton from there.

In the “Project” window, open the “Graphics” folder. Click on the “Library” icon. In the right side of the window, scroll down to the “Buttons – Industrial” display and open it.

This display appears.



Let’s use the style in the lower left hand corner.

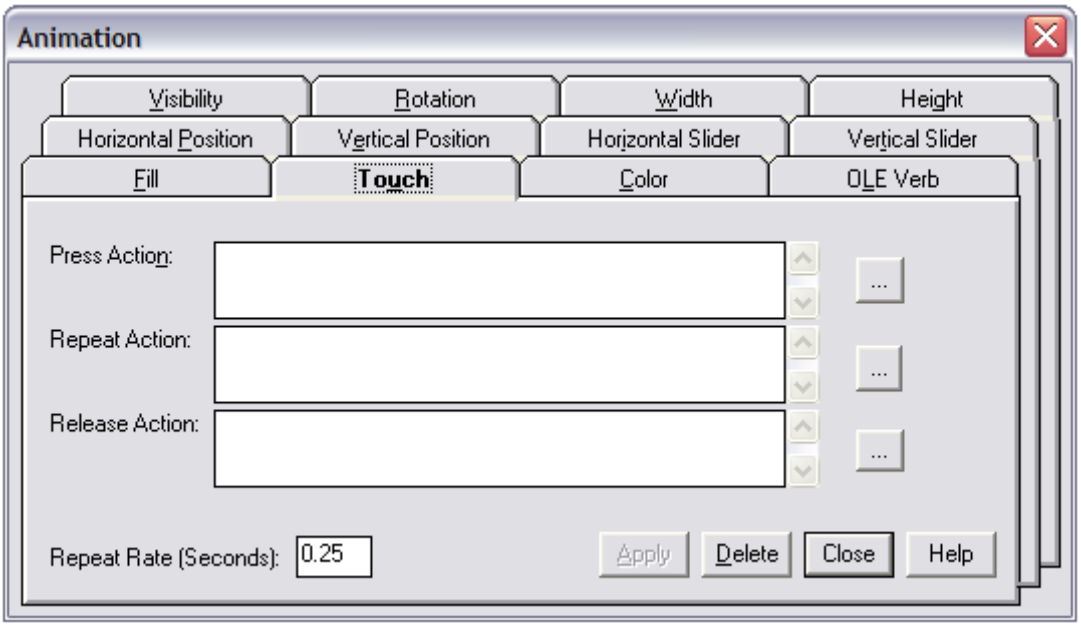
Simply click and drag the “Stop” button onto your “System View” display. We will need to change the color of the button and the text, so the object will have to be ungrouped. Select it and use the top menu (Arrange > Ungroup) to achieve this.

Select the red circle and change it to black, which is the typical standard color of a reset button.

Edit the text to say “RESET”.

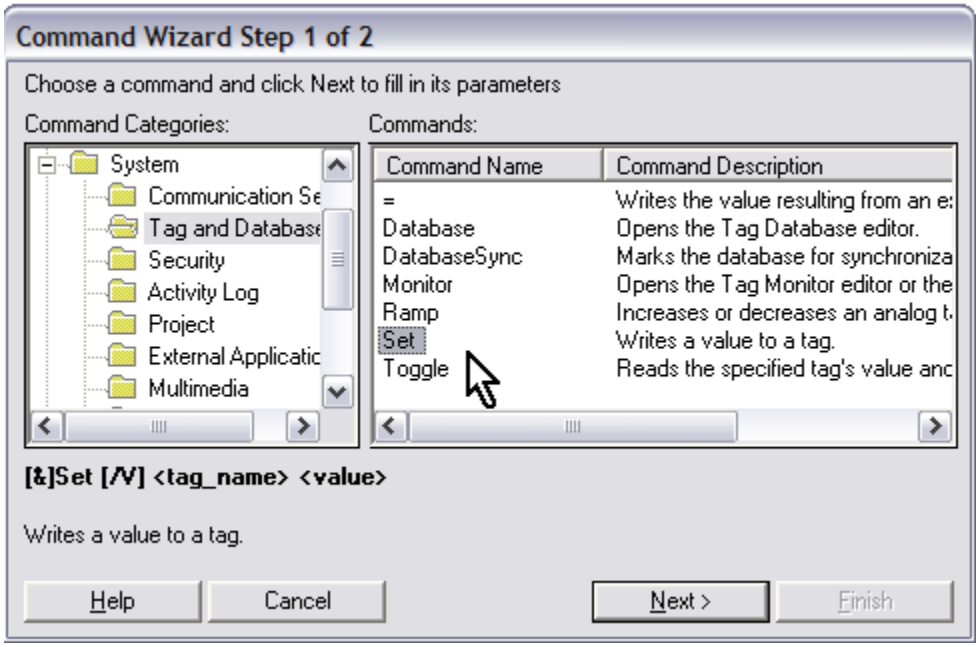
How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

This window appears.



We have the option of a “Press Action”, a “Repeat Action” and a “Release Action”. In this case, all we care about is press and release.

Click on the button with the dots to the right of the “Press Action “ box. The “Command Wizard” window appears. Open the “System” folder. Open the “Tag Database” folder. Click on the “Set” command, as we want to write a value to a tag.

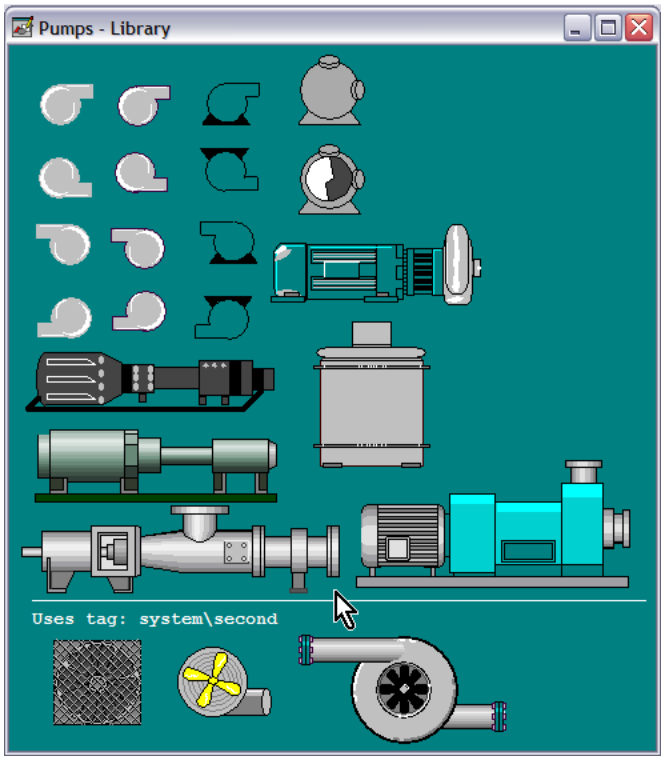


Click next

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Pump Icons

Open the “Pumps” graphic from the standard graphics library.



Select the pump icon that matches the icon shown in the scope and drag it to your display. It is much too large, so use the handles to resize it.

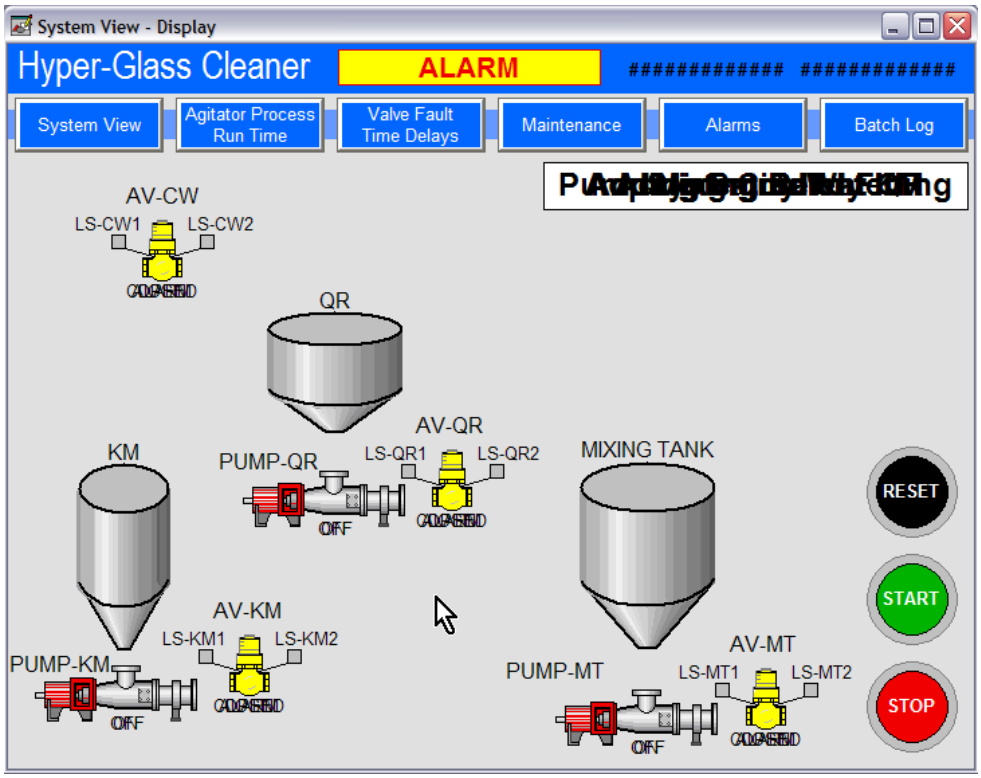
Let’s start by making this the icon for the KM pump (PUMP-KM). Add the identifying text to the pump. Add the text identifying the state of the pump (ON). Copy the graphic to another area of the screen.

We won’t need an alarm state for the pump, as the scope did not request this. We will also not need to have a gray icon, as the pump will only be either on or off.

Change the fill color of the running pump icon (ON) to green. Make a group from the icon and the “ON” text. Animate the visibility if the group to be visible when tag RunKMPumpPUMP-KM is true.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

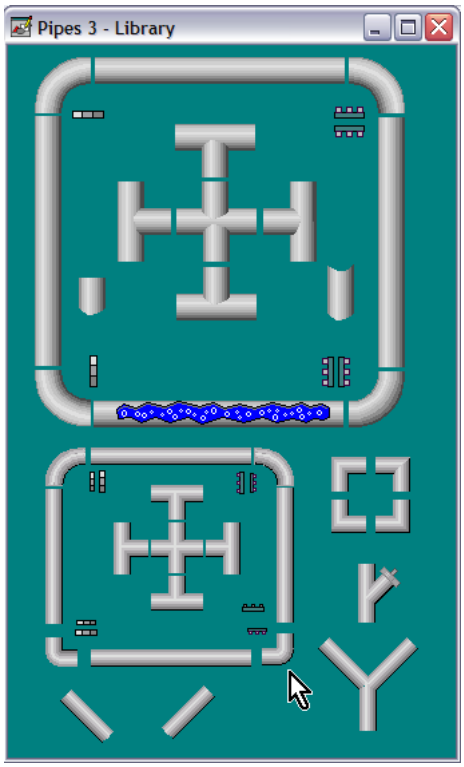
Your screen should now look something like this.



How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Adding Piping

Open the “Pipes 3” graphic from the library.



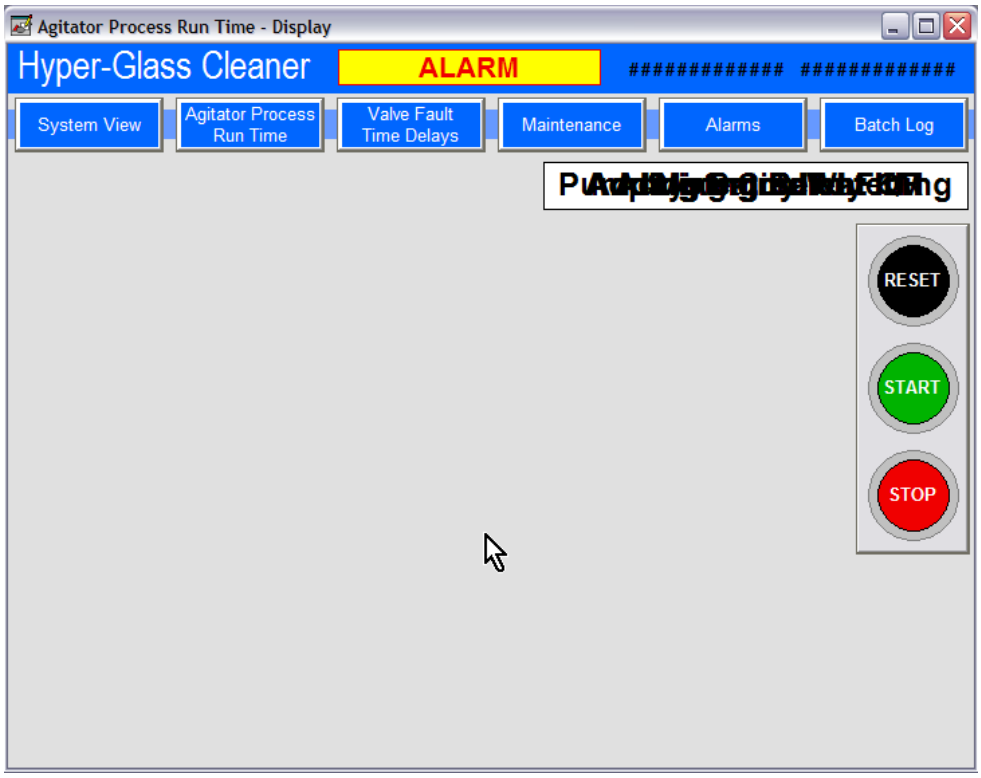
Use the smaller set at the bottom of the graphic. Place the pipes as necessary. You may have to change the length, but don’t try to get the pipes to “mate up” exactly with the other elements. It goes much more quickly if you leave the pipes a bit long and simply send them to the back, behind the element to which you are connecting.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Screen 2 - Agitator Process Run Time Display

The scope says that the “agitator process run time is adjustable from 60 seconds to 360 seconds”.

Open the “Agitator Process Run Time” display. Clean off all of the graphics that applied to the system view so that the display looks like this.



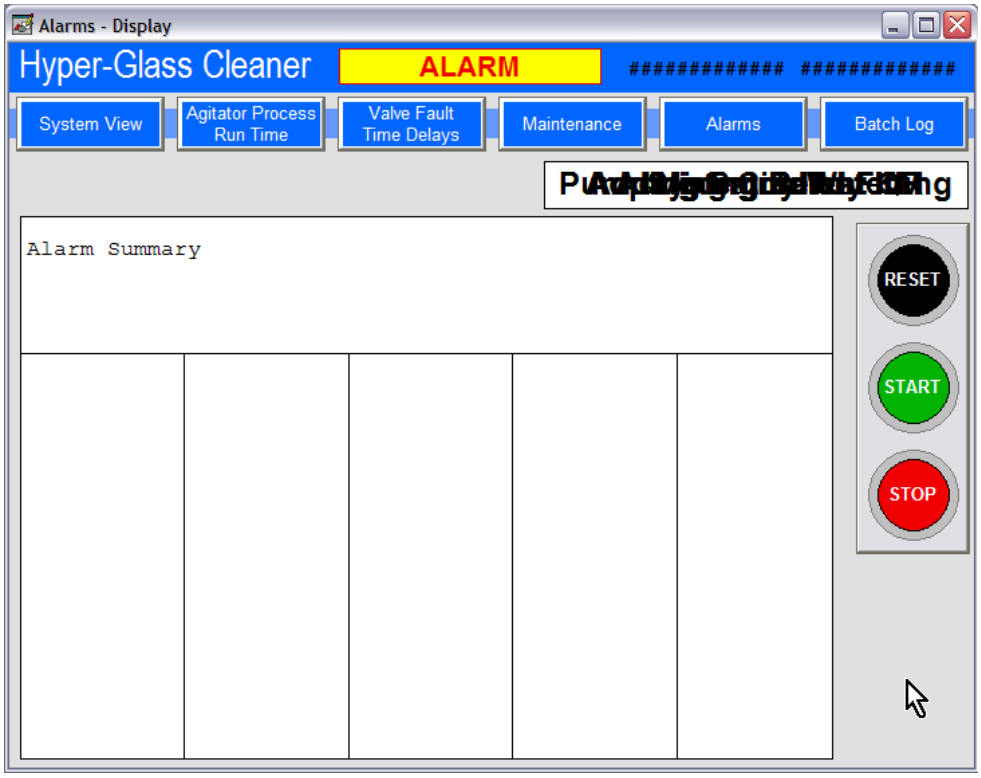
Click on the “Numeric Input” button on the toolbar. Draw a rectangle on the display. When you release the mouse, the “Numeric Input” window appears.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Screen 5 - Alarms Display

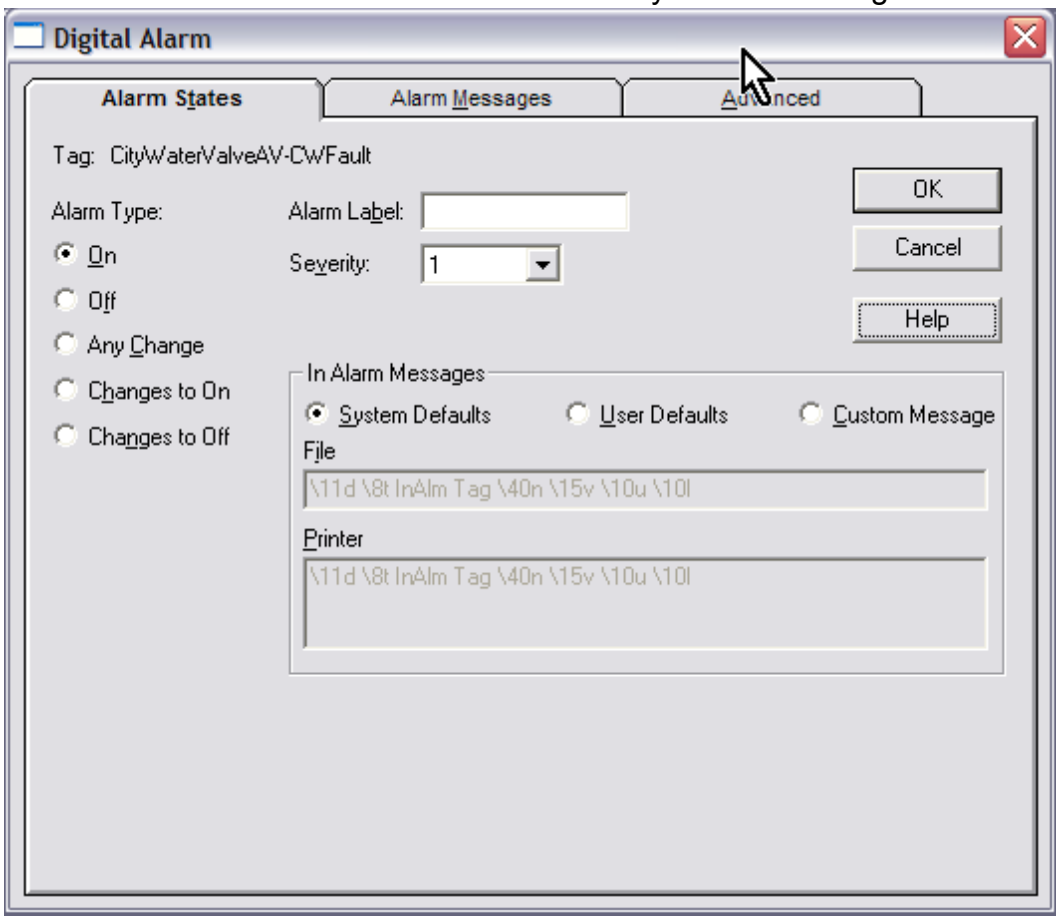
From the “Project” window, open the “Alarms” display.

From the top menu, click “Objects > Advanced Objects > Alarm Summary”. Draw a rectangle that fills the screen as much as possible without overlapping any existing graphic elements.



How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Under “Alarm Type”, we can select the alarm to trigger when the tag turns on, off, changes, changes to on or changes to off. The “changes” types differ from a normal “on” or “off” in that these are considered out-of-alarm immediately after the change of state.



For our purposes, we can simply leave the default setting as they are. Click “OK”.

If you ever need to reconfigure the alarm parameters for a given tag, just click on the “Alarms” button.

Configure the alarm parameters for the following tags, as listed in the scope.

- Valve AV-QR Discrepancy Fault
- Valve AV-KM Discrepancy Fault
- Valve AV-MT Discrepancy Fault
- Mixing Tank Hi Level (above 95% of capacity)
- Emergency Stop Activated

In the case of the “Emergency Stop Activated” alarm, we will use the “E-STOPClearedPB3” tag and set the alarm to trigger when the tag is off.

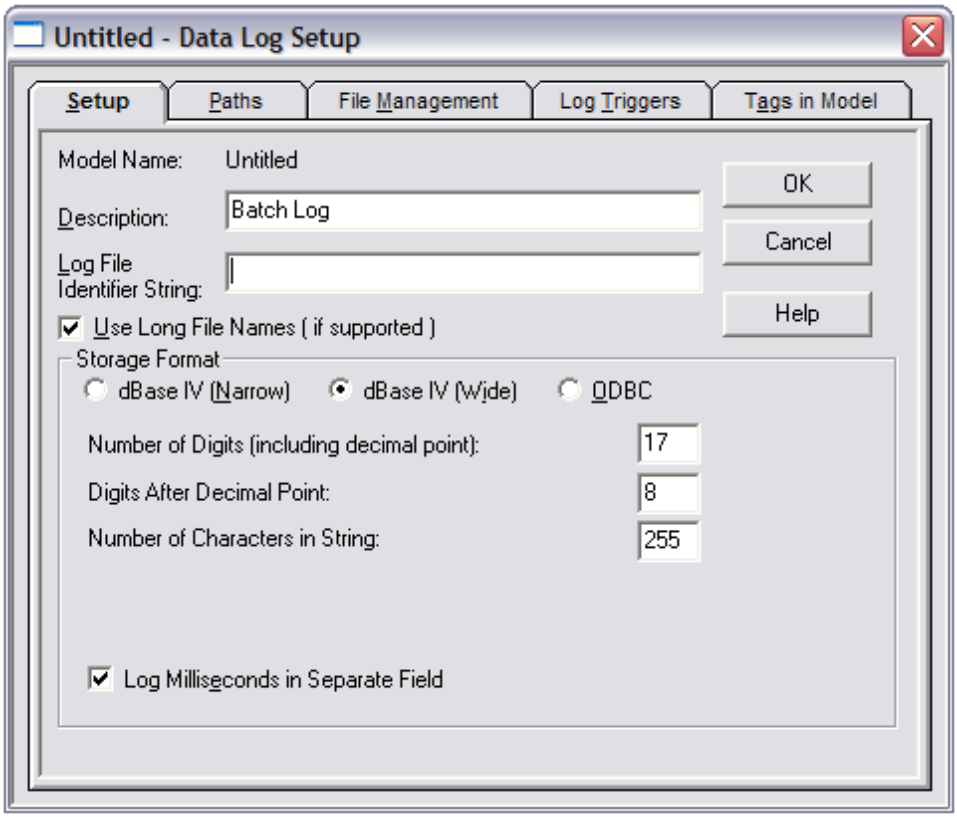
How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Screen 6 - Batch Log display

The scope says that the “completion date and time of each batch is recorded. This information is maintained for a minimum of 180 days”.

The first thing we have to do it setup our data log. In the “Project” window, open the “Data Log” folder and click on “Data Log Setup”.

After this window opens, type “Batch Log” in the description.



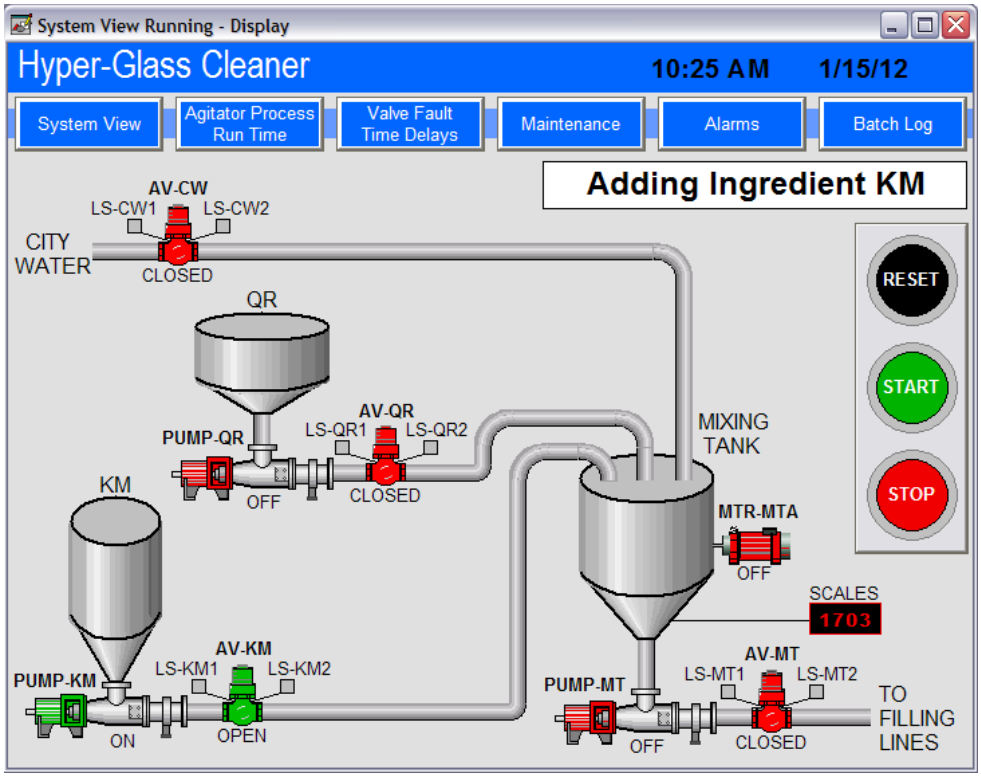
Use the “Path” tab to set the path to your log file.

Click on the “File Management” tab. We want only one file, so choose “Never” under the “Start New Files” section.

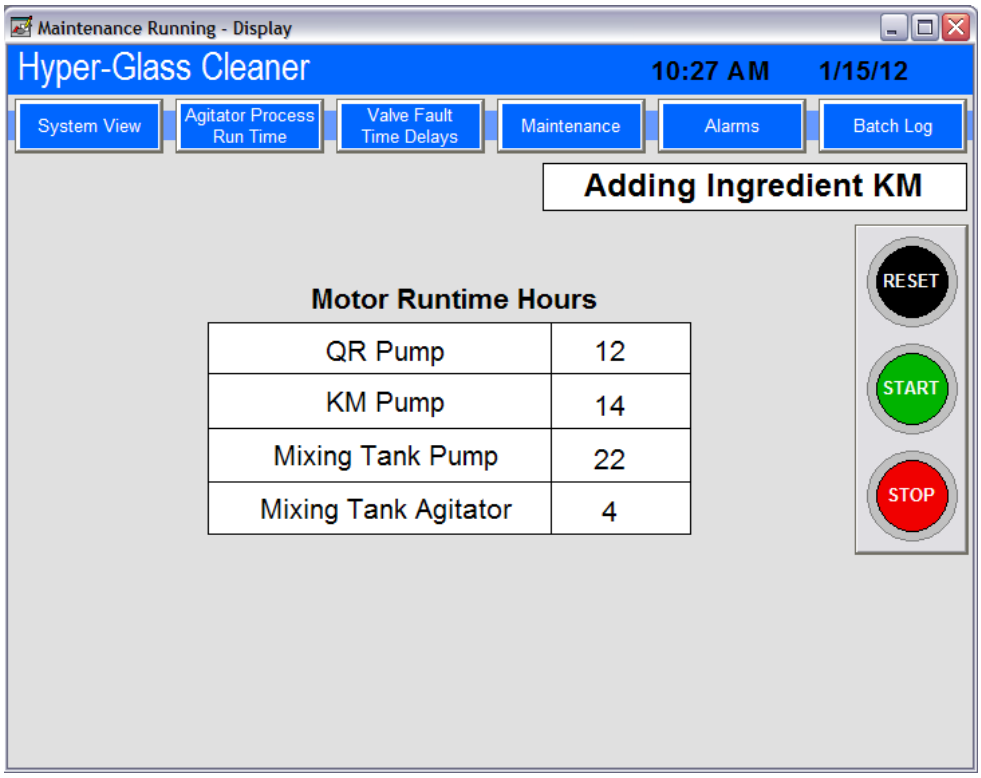
How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

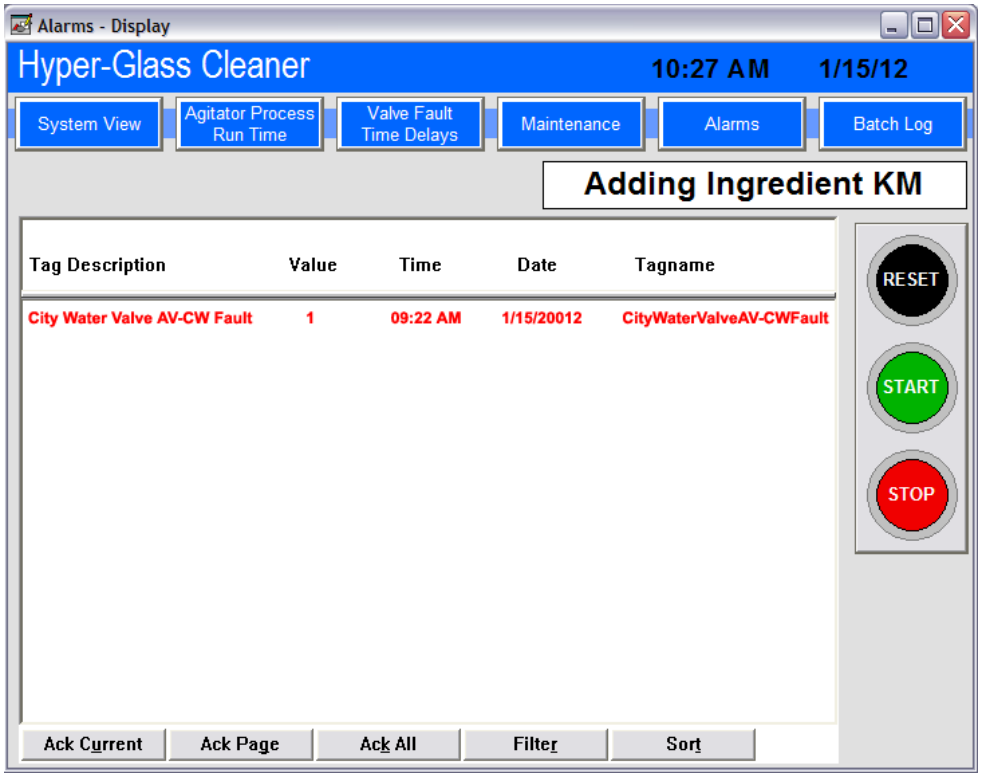
The Final Result

Let's take a look at how the screens look in actual operation.

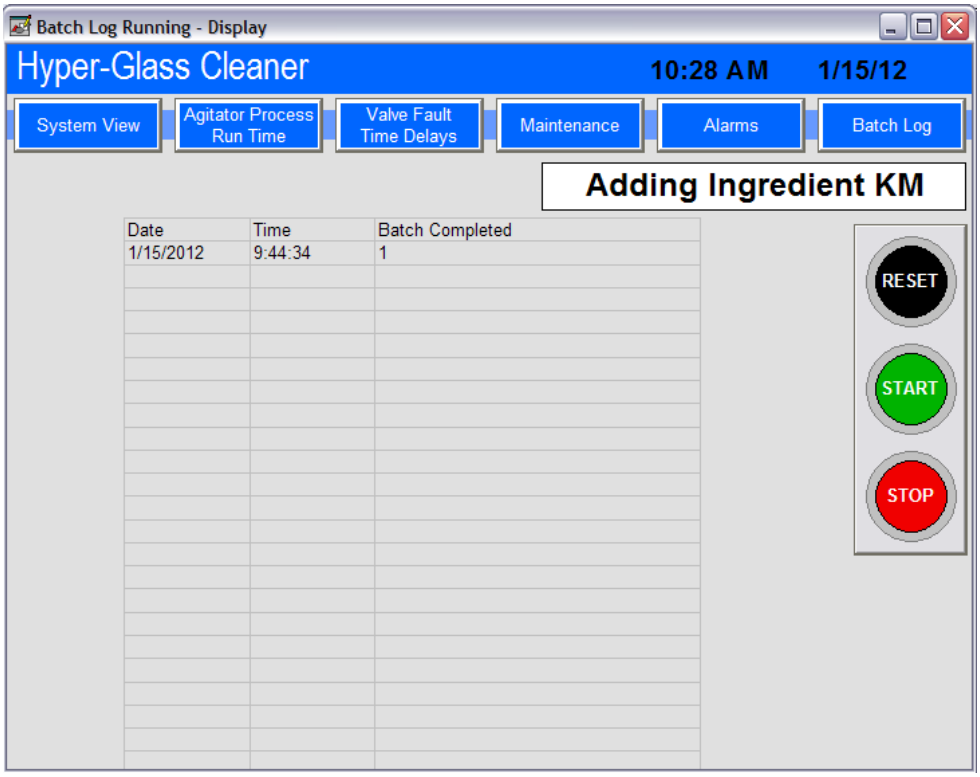


How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com





How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

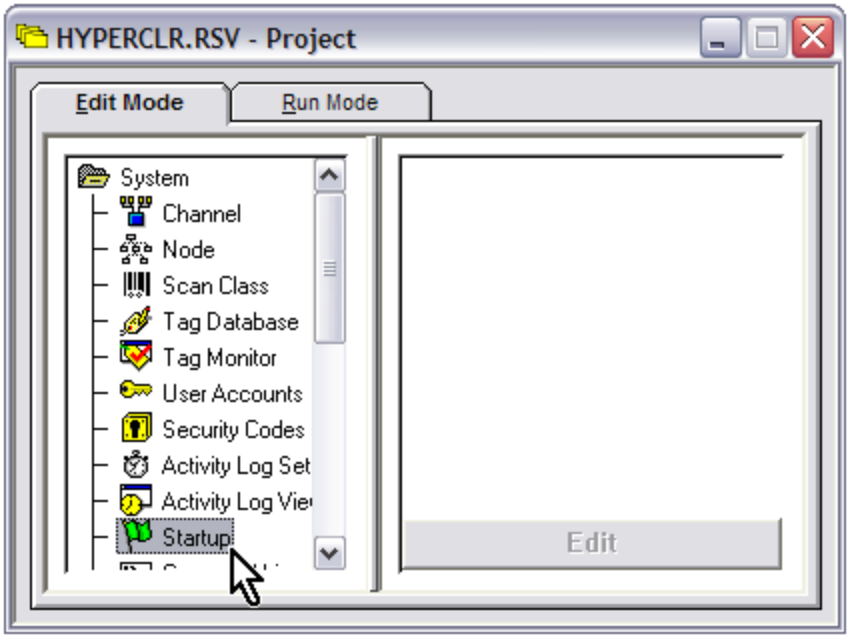


How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Finishing Touches

We still have some configuration changes to make so that RSView will start properly.

In the “Project” window, open the “System” folder.



How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Activity Bar
Check this box to display the activity bar at the bottom of the screen. Again, in our application, this is not necessary.

Minimize Button
Check this box to display a minimize button in the top right hand corner of the window.

Maximize Button
Check this box to display a maximize button in the top right corner of the window.

Control Box
Check this box to display a control menu box in the upper left corner of the window. When this button is clicked it displays the control menu that allows the user to manipulate the window. We don't want to give the operators access to this.

Project Manager
Check this box if you want the "Project Manager" to be displayed at runtime.

Disable

Ctrl-Alt-P (Project Manager)
Check this box to disable the key sequence that will turn on the display of the Project Manager.

*It is important to remember that if the "Project Manager" box is not checked, you can press **Ctrl-Alt-P** to toggle the Project Manager at runtime if you have not disabled the key sequence. Under some conditions, this is the only way to open the "Project Manager" window. It may the only way you can edit the displays or the program configuration.*

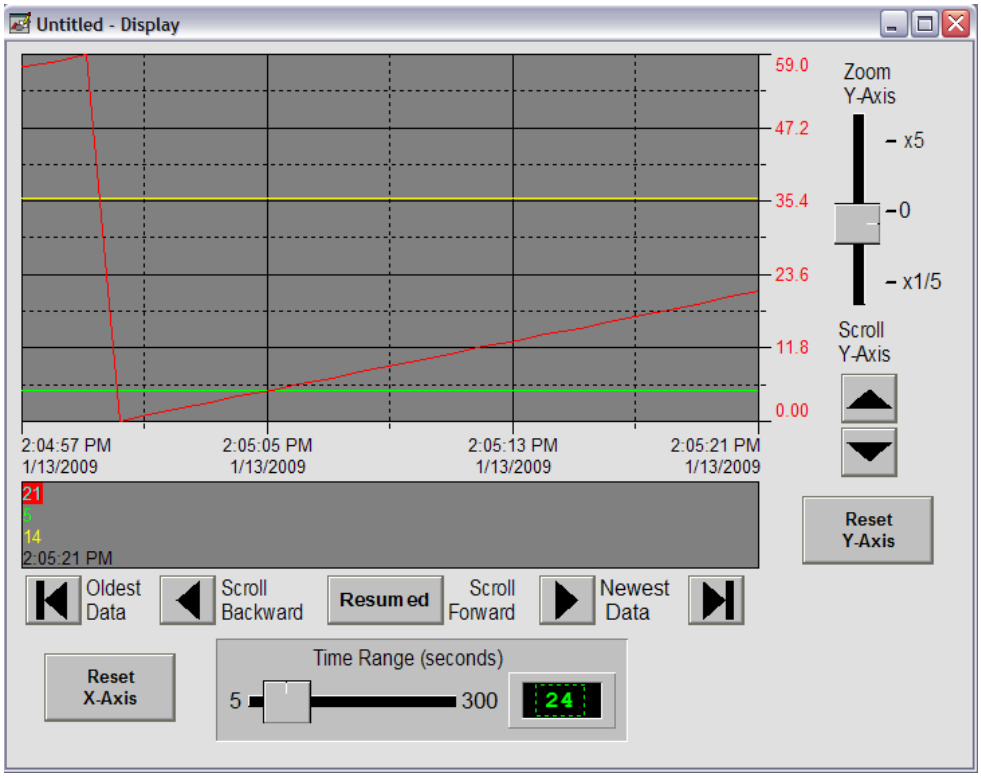
Switch to other Apps
Checking this box will prevent users from switching to another application. Checking this option also disables Ctrl-Alt-Esc, Alt-Tab, Alt-Esc and Alt-F4.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

Trending

RSView provides a lot of flexibility with its trends. You can set up a trend in just about any way you would like.

There is a very elaborate trend available in the “Trends” display of RSView’s library.



This trend has three variables, with the actual values shown in the box just below the graph. As you might expect, the red value corresponds with the red line, the green value with the green line and yellow with yellow.

The unique part of this graph is the array of buttons and arrows below and to the right of the graph. A trend that is created from the menu (“Objects > Advanced Objects > Trend”) would only contain the graph.

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

The example above shows that the trend is graphing seconds, minutes and hours of the system clock. We could add any tag we like, whether analog or digital. Colors, scales and times frames can all be adjusted.

Trends, though, tend to consume a fair amount of computer memory, and may affect the performance of your system.

Tips and Tricks

Leave the Control in the PLC
Keep in mind, that although you can change bits and control the process with RSView, avoid that if you can. If the PC that is running RSView fails, the whole process could be halted.

Copy and Paste First
When you need a new piece of text, or a new graphic such as a rectangle, look to see if you can copy an existing object. It is almost always much faster to copy something and then change it to make it what you need, as opposed to going to the toolbar and "starting from scratch".

Make Sure You Know The Screen Resolution Of The Display On Which The Project Will Be Shown, And That It Is Documented
RSView will adapt to minor changes in screen resolution, but big variations can cause problems.

Don't Expect To See Rapid Changes To Be Visible In RSView
Due to the communication speed, sometimes bits in the PLC turn on and off so quickly that RSView is not able to catch them. Sometimes, there can be a lag of 2 or 3 seconds.

Formulas can be included in expressions
For example, we could animate the color of the text in the "Scales" display of our "System View" graphic.

Put this expression in the "Color" section of the "Animation" window.

LiquidLevelOfMixingTankPercent > 90

Select a color, such as yellow, and the text will change to yellow when the tag value is greater than 90.

Expressions can use logical operators, arithmetic operator and a variety of functions. Here is the list of functions available in RSView:

How to Program RSView32
Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

The Steps to a Successful and Profitable RSView Project

We all want to create a project as accurately and efficiently as possible. If the steps shown below are followed, then an RSView program can be completed with maximum accuracy and minimum cost.

Of course, the real world will throw curveballs and changes to you. As a result, you will have to go back and re-work certain aspects of your design.

Ideally, though, this is how things should work:

1. Get a clear and concise scope that defines the project in as much detail as necessary.

2. If the scope is not clear, write emails and document any clarifications or changes to the scope.

3. Understand the scope fully. Get your head inside the process that you will be representing with RSView.

4. Make sure that the PLC program is finished and that all necessary tags have been defined by the programmer.

5. Create mock-ups of your screens and get client or supervisor approval before you do any real programming or configuration.

6. Test your navigation on your co-workers. Humbly accept their input and make any changes necessary.

7. Test your design on the actual PLC that will be used in the project before you take it into the field.

I wish you the best of luck in your endeavors.

Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com

THE FOLLOWING ARE TRADEMARKS OF ROCKWELL AUTOMATION, INC.

Allen-Bradley®
ControlLogix™
CompactLogix™
MicroLogix™
PanelView™
RSLinx®
RSLogix™
RSView32™
RSLogix™ 5000
SLC™ 500

THE AUTHOR OR THE PUBLISHER OF THIS BOOK IS IN NO WAY AFFILIATED WITH ROCKWELL AUTOMATION, INC.

Disclaimer

THE AUTHOR INTENDS THIS DOCUMENT TO BE ADVISORY ONLY. ITS USE IN INDUSTRY OR TRADE IS STRICTLY VOLUNTARY.

THIS DOCUMENT IS PROVIDED BY THE VENDOR “AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE VENDOR OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Courtesy of Rockwell Automation, Inc.

Copyright © 2013 Modern Media & Automation, LLC
engineer-and-technician.com