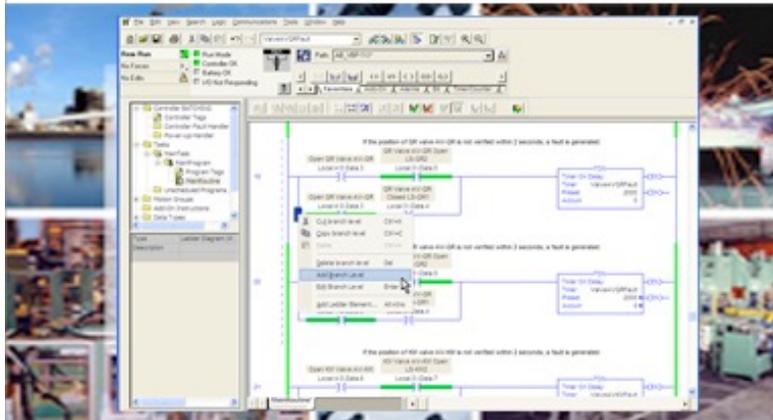


How To Troubleshoot With A PLC



*How to Diagnose and Correct Problems
with RSLogix 500 and RSLogix 5000*

by Neal Babcock

How to Troubleshoot With A PLC

*How To Diagnosis
And Correct
Problems With
RSLogix 500 And
RSLogix 5000*

By Neal Babcock

Industrial Automation Series

engineer-and-technician.com

Contents

| | |
|--|-----|
| Introduction | 1 |
| PLCs | 3 |
| The Allen-Bradley SLC Family of PLCs | 4 |
| Hardware | 4 |
| SLC Rack | 4 |
| SLC Power Supply | 4 |
| SLC Processors | 5 |
| SLC I/O Modules | 5 |
| RSLogix 500 Ladder Logic | 5 |
| The Dialect of SLC PLCs | 6 |
| Connecting To The SLC And Going Online | 9 |
| The Allen-Bradley Family of ControlLogix PLCs | 17 |
| RSLogix 5000 Ladder Logic | 17 |
| Connecting To The ControlLogix PLC And Going Online | 21 |
| The Overall Method | 23 |
| Developing A Clear Goal And A Concise Plan | 25 |
| The 6 Steps to Creating a Clear Goal | 25 |
| The 6 Steps to Developing a Concise Plan | 28 |
| Troubleshooting And Diagnosing Existing Equipment | 31 |
| The Automated Drill Press | 31 |
| Evaluate the Symptoms | 38 |
| Troubleshooting the Logic | 39 |
| Searching the Program | 42 |
| Upgrading An Existing Machine With New Components Or Programming | 52 |
| Know How to Get Back to Where You Started | 52 |
| Always Know How To Return A Program To Its Original State | 52 |
| Modifying Or Enhancing Existing Equipment | 54 |
| The Automatic Batching System | 54 |
| Modifying an Existing or Enhancing a Program | 54 |
| Confirming That The System Is Running Properly | 62 |
| What If You Don't Have The Exact Replacement Part? | 62 |
| Troubleshooting With RSLogix 5000 | 65 |
| Finding The Problem | 68 |
| Investigating An Analog Input | 71 |
| Finding And Correcting Intermittent Problems | 73 |
| Tracking Down Intermittent Problems | 73 |
| Modifying A Program To Trap Faults | 77 |
| Accepting Rung Edits In RSLogix 5000 | 84 |
| Forcing an Output | 87 |
| You Have Found The Problem – What Are The Options? | 92 |
| Troubleshooting Tips | 101 |

troubleshoot

verb [intrans.] [usu. as n.] (**troubleshooting**)

- solve serious problems for a company or other organization.
- trace and correct faults in a mechanical or electronic system.

New Oxford American Dictionary

Troubleshooting is a form of problem solving. It is a systematic search for the source of the problem that is keeping a device, machine or system from running.

Troubleshooting skills are needed to repair, improve and maintain complex systems. These skills are used to determine possible causes of the problem, and then the process of elimination is utilized to find the real cause.

The last stage of troubleshooting confirms that the solution restores the device, machine or system to a working state.

The purpose of this book is to teach you how to quickly and accurately troubleshoot a system that is controlled by a PLC.

If you are called upon to troubleshoot a problem, you may not have had any involvement in the in the original programming of the system, or machine. You may not have ever seen this machine before.

Still, you are responsible for getting it running again. Suddenly, the pressure is on and you are on the hot seat.

There are two primary aspects of troubleshooting a system; the technical aspect, which is the knowledge of the equipment and the procedural aspect, which is the knowledge of what steps to take to efficiently troubleshoot a system.

A working knowledge how to effectively implement both aspects is essential to effective troubleshooting.

Whether it is a new system that doesn't work yet or an old system that has stopped working, getting it to run involves troubleshooting.

The scenarios presented in this book are based on actual events.

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 troubleshoot a PLC controlled system.

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

TIP PLC program. It is available from Modern Media. Visit engineer-and-technician.com if you would like to learn more about this book.

RSLogix 500 is the software used to program the SLC family of PLCs.

RSLogix 5000 is used for the ControlLogix PLCs.

Though the general techniques we are going to cover are applicable to PLCs from any manufacturer, we are going to concentrate mostly on Allen-Bradley PLCs in the book, specifically the SLC 500s and ControlLogix PLCs. The software that these PLCs use is RSLogix 500 and RSLogix 5000, respectively.

We will provide a cursory explanation of both versions of RSLogix. Modern Media provides ebooks explaining this software in detail at engineer-and-technician.com.

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. If you can, establish a relationship with your local technical support team of the manufacturer of the PLCs used in your applications. Get to know them *before* you run into a problem. They are typically very helpful and they can save you hours of frustration.

Many manufacturers 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.

SLC Processors

There are five SLC 500 processors available:

SLC 5/01
SLC 5/02
SLC 5/03
SLC 5/04
SLC 5/05

The 5/01 is the most basic processor, with each succeeding model having more capabilities. The most important difference is found in the SLC 5/05, which has the capability of Ethernet communications.

SLC I/O Modules

There are an incredible amount of I/O (input/output) modules available for the Allen-Bradley SLC family. There are 4-20mA and 0-10VDC analog modules. There are digital (also known as discrete) modules that work in a variety of voltage configurations and capacities.

RSLogix 500 Ladder Logic

Before we open RSLogix 500 and start troubleshooting, 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, this information 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 SLC 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.

Allen-Bradley PLCs, like many computers, always start with 0.

By the way, don't get the capital "O's" confused with zeroes.

RUNG - A section of the PLC ladder program that terminates in an output function of some type. Just like in an electrical ladder diagram, a rung has some type of output that is turned on or turned off by the preceding entities in the rung. The first rung in a ladder program is always 0000.

HARDWIRED INPUT - a physical connection to the PLC from an input device (switch or sensor, etc.).

Allen-Bradley uses the capital letter "I" to designate a hardwired input. An address that describes an input on an SLC 500 is I:4/0.

Similar to the output structure,

I:4/0 means that it is a physical input.

I:4/0 means that it uses Slot 4 (the 5th slot in the rack).

I:4/0 means that it is the first input on the card.

Don't get the capital "I's" confused with ones.

HARDWIRED OUTPUT - a physical connection from the PLC to an output device (relay or pilot light, etc.) As was said above, an address that references an output of an SLC 500 is O:5/0.

INTERNAL COIL

This is a programmable bit used to simulate a relay within the PLC. The internal coil has no connection to the outside world. It does not connect to an output card. Internal coils are used to store information. The "contacts" of this "relay" can then be used multiple times in other parts of the program.

In RSLogix, the "B3" (binary) file is commonly used for all the internal coils. There are many other words in other files that have bits you can use as internal coils, but we are going to stick with the B3 file for our applications.

B3:0/0 means that it references an internal Binary file

B3:0/0 means that it uses the first word in the table

B3:0/0 means that it is the first bit in the word.

Note that, unlike the Output and Input files, you have to use the file number in the address. In this case, the default file number is 3.

TIMER

A timer is a programmable instruction that lets you turn on or turn off bits after a preset time.

The two  primary types of timers are TON for "timer on delay" and TOF for "timer off delay".

Timers in A-B SLC and MicroLogix processors use file 4 for their timers.

T4:0 means that it references an internal Timer file

T4:0 means that it uses the first timer in the table

WARNING

There are dangerous voltages present on terminals of the PLC. Follow all warnings and instructions from the Allen-Bradley manual for connecting a laptop to the PLC. If you are not familiar with the hazards and the potential dangers of these voltages please **STOP RIGHT NOW**. Consult a trained professional who is able to assist you.

Select the SLC in the left window and click “OK”. The communications window will close.

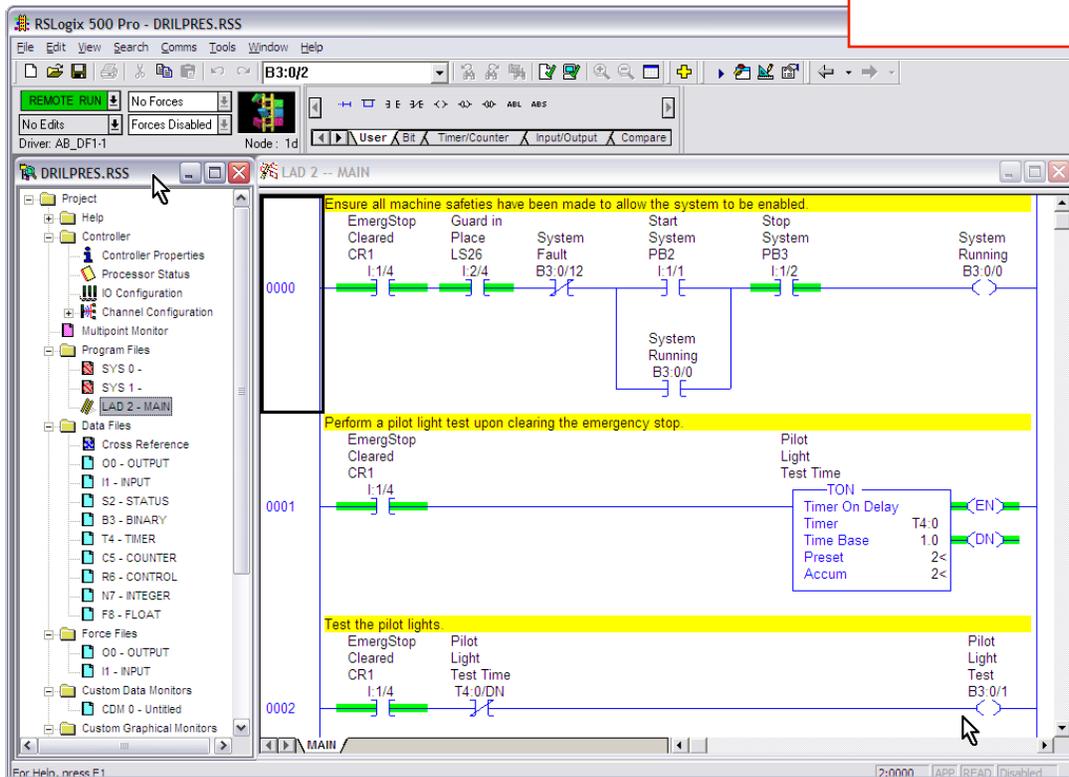
In the RSLogix 500 main screen, choose File. Navigate to and select the program DRILPRES.RSS.



If things go well, you will be able to go online right away.

You will now see a screen like this. Notice that the ladder icon is animated to indicate you are online.

Test your E-Stop circuit thoroughly, even if you are working on a system that has been running for months or years. Be ready to hit the E-Stop button if you see a hazardous situation developing.



Notice the green dropdown in the upper left that says “Remote Run”. Allen-Bradley SLCs have a key switch located on the front of the processor.

The key switch in the PLC has 3 positions: RUN (run mode), REM (remote run mode, meaning that the mode can be selected from RSLogix) and PROG (program mode).

If the switch is placed in RUN, the PLC will run, but no changes can be made to the program.

The Allen-Bradley Family of ControlLogix PLCs

Hardware

Like the SLCs, the minimum requirements are a processor, a power supply, a rack and some I/O modules.

If you will be dealing with a ControlLogix or CompactLogix system, it is recommended that you spend some time at rockwellautomation.com

RSLogix 5000 Ladder Logic

RSLogix 500 and RSLogix 5000 share a few common characteristics and conventions. Some of the information here is similar, if not identical, to the information presented in the RSLogix 500 section.

INSTRUCTION – RSLogix 5000's Relay Ladder Logic 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.

RUNG - A section of the PLC ladder program that terminates in an output function of some type. Just like in an electrical ladder diagram, a rung has some type of output that is turned on or turned off by the preceding entities in the rung. The first rung in a ladder program is always 0.

HARDWIRED INPUT - a physical connection to the PLC from an input device (switch or sensor, etc.).

RSLogix 5000 defines the address of the input, based on the input cards that you configure.

Here is an example of a hardwired input:

A counter is a programmable instruction that lets you turn on or turn off bits after a preset count has been reached.

There are different types of counters available in RSLogix, but the CTU (counter up) instruction covers everything we will talk about here.

Counters in RSLogix 5000 use tag names for identification.

--] [-- **Normally Open Contact**

When used with a hardwired input, this instruction is off until there is a voltage applied to the input. The bit address then goes high, or on, and the instruction becomes “true.” It works the same way when it has the same address as an internal coil, except that the coil must be turned on by logic in the program.

Allen-Bradley calls these normally open contacts “XIC”, or “eXamine If Closed” instructions.

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 “eXamine If Open” 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 “**Output Energize**”.

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.

Connecting To The ControlLogix PLC And Going Online

Connecting to a ControlLogix processor is done through Ethernet, the same way you connect a PC across a network.

Typically, most ControlLogix PLCs are connected through an Ethernet switch. You simply plug your computer into an open port on the switch.

Click on:

Communications > Who Active

Navigate to the PLC to which you would like to connect.

To be frank, this is where many people have problems. Sometimes connecting to a PLC is as easy as finding an Internet site in your browser. Other times, it can be an incredibly frustrating experience, due to no fault of your own.

Do what you can to prepare yourself by asking others about connecting to this particular PLC, making sure your computer can connect to a similar PLC, and so on.

Anybody who has ever connected to a PLC has had problems; they will understand your position.

There was one instance where the firmware in a particular PLC was not compatible with the version of RSLogix I was running in my laptop. It took a call to Rockwell's Tech Support to sort out the problem. They were very helpful, and it wasn't long before I was connected.

The Overall Method

Any successful troubleshooting operation, not matter how large or small, always has two components; a **clear goal** and a **concise plan**.

Let's take a very simple example. Your wife calls you from the living room and tells you that the light bulb in her favorite lamp has burned out and she wants you to replace it. Being a seasoned troubleshooter, you immediately visualize a clear goal and develop a concise plan.

Even though she said she wants you to replace the bulb, *the real goal is to get the light to work*. Simply replacing the bulb may not achieve the goal.

In the interest of domestic harmony, there is really no reason to discuss that part with her. You are using discretion, but also realizing that the light bulb may actually be fine, and that the real problem may lie elsewhere.

You might be thinking right now that I am stating the obvious, and in this case, that may be true. On a factory floor, however, be aware that people sometimes develop a plan for you that might not be the best approach.

On your way to the living room, you are thinking of possible causes for the malfunction.

- Is the lamp plugged in?
- Is the switch turned to the correct position?
- Is the bulb secure in the lamp socket?
- Is there a wall switch that switches power to the outlet that the lamp uses?
- Could the switch in the lamp be broken?
- When was that bulb replaced?
- Do you have a new bulb of the same wattage?

Your first step it to verify that the lamp is plugged in. Given that it is, you operate the switch, yet the bulb still does not illuminate. You reach into the lamp to verify that the bulb is secure and find that it is tight in the socket. Looking across the room, you see a

wall switch and remember that this is a switched outlet. Turning on the switch now causes the bulb to be illuminated.

Your wife is happy and domestic bliss is maintained, even though her stated goal of having you replace the bulb was not met (let's not make a big deal out of that, though).

Suppose, instead of using good troubleshooting techniques, you simply followed her instructions.

You open your utility room door and dig around until you find a package of light bulbs. You grab one and walk out to the living room. As you unscrew the bulb from the lamp, you realize that it is a 75-watt bulb, and that the bulb you brought from the utility room is a 40-watt bulb. You replace it anyway, only to find that it still does not work.

Your wife is still not happy, even though you did what she asked you to do. Domestic harmony is threatened as patience is wearing thin.

Finally, you realize that *the real goal is to get the light to work*.

As a last resort, you begrudgingly follow proper troubleshooting techniques. You discover that the wall switch had been mistakenly turned off. The lamp now works, but since you already replaced the original 75-watt bulb with a 40-watt bulb, the light is insufficient and your wife is still not happy.

You turn off the lamp, and wait until the bulb is cool enough to change without burning your fingers. Finally, you are able to change out the bulb and your wife is *reasonably* happy.

You, though, feel a little frustrated in realizing that you spent 5 minutes to fix a problem that should have taken 30 seconds. You show the greatest wisdom, however, by not questioning your wife's ability to operate the wall switch that would have powered the lamp in the first place . . .

This is a very simple illustration, but it is important that you understand the importance of having a clear goal and a concise plan.

Developing A Clear Goal And A Concise Plan

Troubleshooting in engineering applications typically fall in one of three scenarios:

1. Repairing existing equipment (a machine that was functioning previously)
2. Upgrading an existing machine with new components or programming
3. A machine start-up

Repairing existing equipment requires you to take a slightly different approach than what you would use with a start-up of new equipment. We'll get into the specifics later on, but all three scenarios still require a goal and a plan.

Keep in mind that a lengthy meeting is not always required to create a clear goal. With practice, sometimes you can get all the information you need as you are walking to the machine.

The underlying secret to all this is effective communication.

The 6 Steps to Creating a Clear Goal

Understand The Problem

Set The Primary Goal

Identify The Secondary Goal If The Primary Goal Cannot Be Met

Obtain The History Of The Equipment

Verify The Timeframe

Evaluate What Resources (Personnel, Equipment And Parts) Are Available

Verify The Timeframe

How long do you have to get the equipment operational again? In this example, there might be enough water in the tower to supply the community for 24 hours. If Pump 7 continues to run, there really is no crisis. If Pump 7 fails, though, the clock begins ticking.

On a factory manufacturing line, if a machine stops, usually the whole line stops. Production is halted until the machine is repaired. The time frame for repair is, as they say, ASAP.

Evaluate What Resources (Personnel, Equipment And Parts) Are Available

What maintenance personnel are familiar with the pump?

Are they available?

What spare parts do you have for the pump?

Is there a spare pump available in the warehouse or at the vendor?

Who is the vendor?

Can the vendor be contacted, if necessary?

The 6 Steps to Developing a Concise Plan

Ask the Right Questions

Objectively Assess The Costs

Establish The Timeframe

Coordinate Your Resources

Address Personnel Safety And Machine Safety Issues

Plan All The Way To The End



Be extremely careful.

Take extra measures by removing power, if possible.

Always look around to make sure there is no one near the equipment when you turn it on.

Always let people know when are planning to start a piece of equipment.

Ask the Right Questions

Has anything changed, such as the installation of a new component or programming changes?

What type of PLC is used to control the machine?

Objectively Assess The Costs

Sometimes, you are simply not concerned with costs. In our example of a failed Pump 6, your manager may be prepared, if necessary, to order a new pump and have it installed immediately, whatever the expense.

In most cases, you will want to keep costs in mind. I remember a situation where I was instructed to write a program in a PLC to maintain tension on a web-handling system. The “web”, in this case, was vinyl that was used in automobile interiors.

Troubleshooting And Diagnosing Existing Equipment

TROUBLESHOOTING OPERATION #1

AUTOMATED DRILL PRESS DOWN

The Automated Drill Press

The best way to learn how to troubleshoot is to look at a real world example. However, before you can do any troubleshooting, you must have a clear understanding of how the machine is supposed to work. Let's say a furniture manufacturer has a machine that drills a 3/8" hole in a certain spot on a piece of wood. The entire process is automated. The isometric drawing below shows the main components of the machine.

Modifying Or Enhancing Existing Equipment

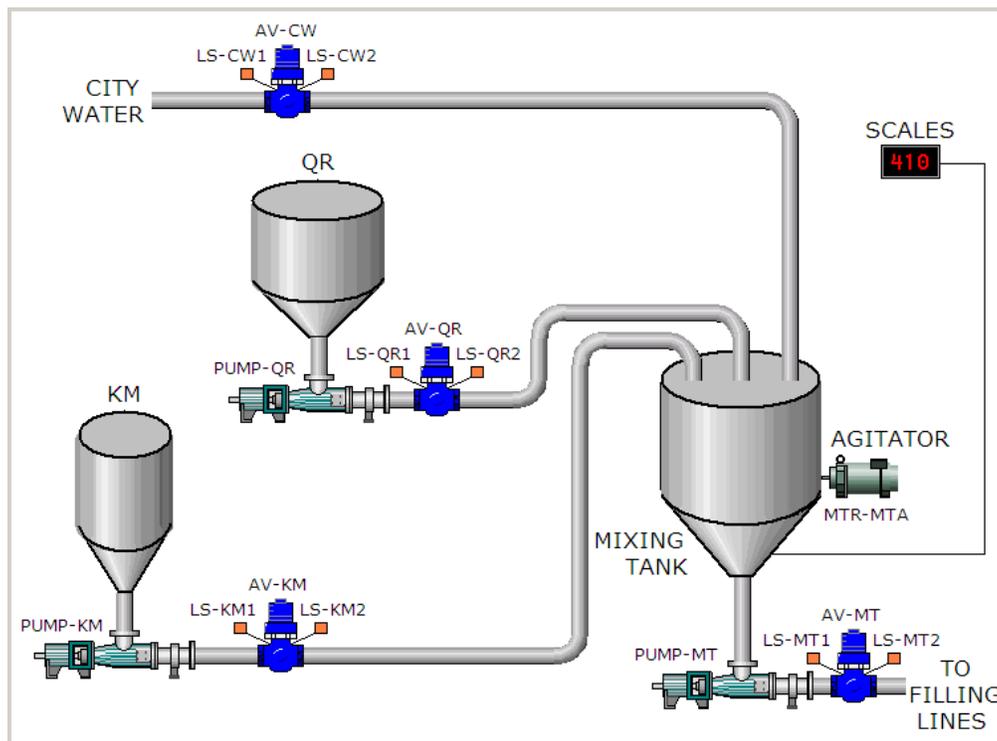
TROUBLESHOOTING OPERATION #2

PREVENT A PUMP FROM RUNNING IF THE RECEIVING TANK IS FULL

The Automatic Batching System

Modifying an Existing or Enhancing a Program

Let's say we have batching system that looks like this.



Three ingredients (city water, ingredient QR and ingredient KM) are added in specified amounts by weight to the Mixing Tank. After all the ingredients have been added to the Mixing Tank, the mixture is blended by running the agitator for a given time. When the

Troubleshooting With RSLogix 5000

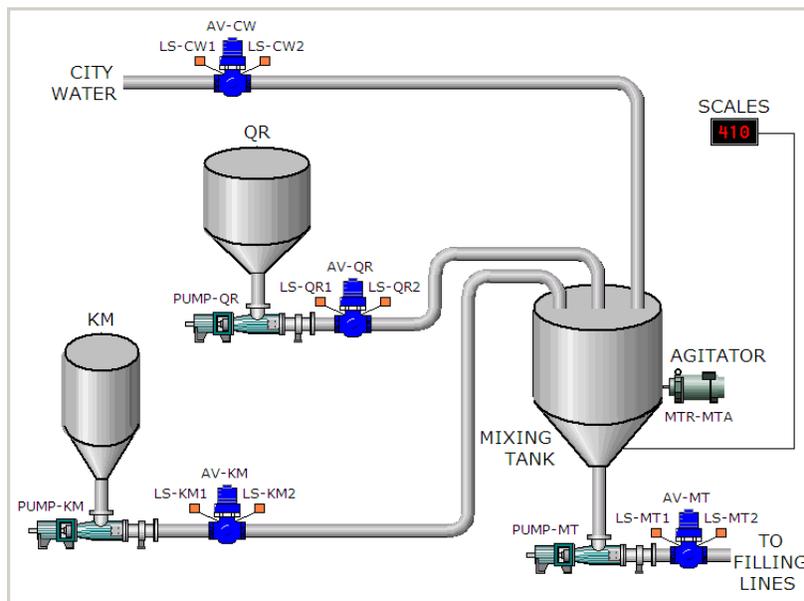
TROUBLESHOOTING OPERATION #3

BATCHING SYSTEM WON'T PUMP FINISHED PRODUCT

Let's consider a batching system just like the one we worked on earlier, except that the PLC is an Allen-Bradley ControlLogix. The software we use to program a ControlLogix PLC is Rockwell's RSLogix 5000.

The problem is that the system is not working. A completed batch was being pumped to the filling lines when the pumped stopped and the "System Fault" pilot light came on. Pressing the "Start" or "Stop" buttons has no effect.

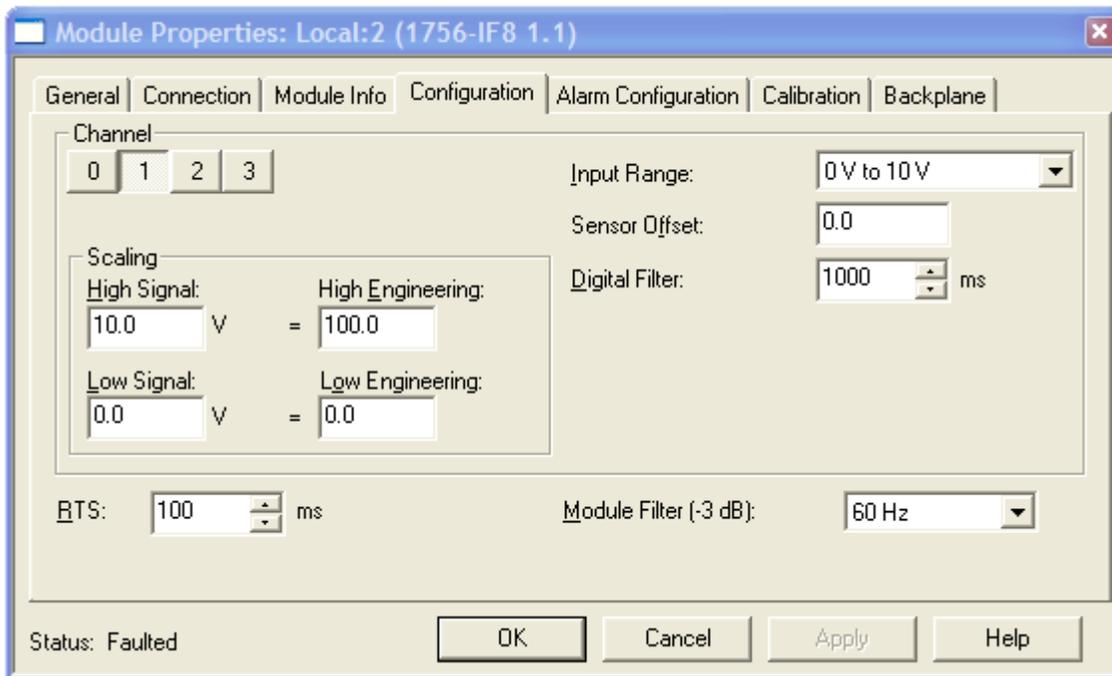
Shown below is a graphical layout of the system. In fact, apart from the program, this is all the information you have.



Investigating An Analog Input

In the Controller Organizer, double-click on "I/O Configuration". On this PLC, slot 2 has a 1756-IF8 card. Right-click on slot 2 and choose "Properties". The "Module Properties: Local:2 (1756-IF8 1.1)" window appears. Click on the "Configuration" tab. Click on the "Channel 1" button.

The configuration is shown like this.



Now we see that the input range on this channel is 0 to 10 volts. The engineering units applied to the scaling suggests that the level in the tank is shown as a percentage. That is, 0 volts equals 0% and 10 volts equals 100%.

Let's go back to Rung 22.

Finding And Correcting Intermittent Problems

TROUBLESHOOTING OPERATION #4

**BATCHING SYSTEM INTERMITTENTLY
STOPS**

Tracking Down Intermittent Problems

The problem is that the system sometimes stops in the middle of a batch. According to the operator, it usually occurs just after ingredient QR is added. The “System Fault” pilot light comes on, but there is no other indication of a problem. Resetting the system by pressing the “Stop” button allows the system to be started again, and it completes the batch normally.

After talking with the operator, you ask him to call you the next time it happens. You also ask him to not reset the system.

So, the problem has occurred again, he called you and you can go online.

Since the system generated a fault and the “System Fault” pilot light is on, let’s go to the fault section of the program.

Rung 22 looks like this.