Email PLC Quiz #121 – Questions

Greetings ...

This edition of our **Email PLC Quiz** contains questions intended for technicians at a **Beginner Level.** As always, the primary objective of the quiz is to cover useful skills for technicians who troubleshoot systems controlled by Allen-Bradley PLCs.

PLEASE NOTE: If you would rather not receive more Email PLC Quizzes like this one, just reply to this email with the word "remove" in the subject line. On the other hand, if you know someone who might find this type of information useful, please feel free to forward this email to them.

IMPORTANT: Due to size and bandwidth considerations, this edition of the PLC Quiz is being sent as an email attachment. If you have trouble opening the attachment, you may download the entire file (in an easy to print PDF format) from our website at www.ronbeaufort.com - look in the "Sample Lessons" section. Answers to the quiz are also available in the same section as a separate file. Absolutely no registration or visitor information is ever required for access to our website.

Also, if you'd like to discuss the information contained in any of our quizzes, please feel free to contact us. We'll be glad to answer any questions that you might have.

Please keep in mind that this material is intended only for use with the PLC-5, the SLC-500, MicroLogix-1000, MicroLogix-1500 and the ControlLogix families of Allen-Bradley PLC processors. You should also keep in mind that there may be certain important differences in operation between these various processor families. For example, information pertaining to a PLC-5 system might not be directly applicable to the SLC-500 or to the ControlLogix platforms. In simple terms, all Allen-Bradley processors do NOT function in exactly the same way.

We'd also like to say "thank you" to all of you who have contacted us and asked for future editions of our PLC Quizzes – and who have recommended your friends and associates to be added to our email list. Based on all of the comments we've received so far, it seems that we're meeting our goal of making our quizzes: (1) entertaining, (2) thought-provoking, and (3) educational. We've also been pleased to hear about the "spirited" debates over our previous editions that have taken place around the maintenance shops. We'd like to say a special "thank you" for all of your excellent suggestions for topics to be covered in our future editions. We'll definitely try to get around to those in the months ahead.

Good luck on the quiz ...

Beginner Level Quiz #121 - “Surprises at Go-To-Run”

Before we get started, here’s a brief preview of the concepts that will be covered in this edition of the Email PLC Quiz.

Suppose that we have two slightly different PLC programming methods to consider. Suppose that both methods give identical results when tested under normal conditions – and suppose that each method works perfectly in the plant’s day-to-day operations.

But now suppose that during a maintenance operation the processor is changed from Run Mode - to Program Mode – and then back to Run Mode again. Or suppose that the system goes through a power cycle – either because of an intentional shutdown, or because of a plant-wide power failure. Or suppose that the power doesn’t fail completely – but only “flickers” for a fraction of a second. All of these different scenarios can be considered “Go-To-Run” situations - and depending on how the PLC’s program has been written, each can give surprisingly different results.

Now suppose that further complications are added to the mix. Consider the sensors and switches and other input devices in the field. Were they ON when we left the Run Mode – or were they OFF? Were they ON when we went back to the Run Mode – or were they OFF? Consider the power supply for the PLC. Is it AC - or is it DC? Consider the power supply for the input and output devices in the field. Is it AC - or is it DC? Once again, depending on how the program is written, all of these different conditions can cause surprisingly different results at “Go-To-Run” time.

Now suppose that we add even more complications – by considering the type of PLC platform being used. Even staying entirely within the currently popular Allen-Bradley brand lines, we have the PLC-5, the SLC-500, the MicroLogix-1000, the MicroLogix-1500, and the ControlLogix platforms to consider. Each of these, in one way or another, can give surprisingly different results from the others at “Go-To-Run” time.

And remember – in each of these cases we’re talking about programming methods which have been “tested and proven” to work perfectly in normal day-to-day operations. The “surprises” only surface when the system is put through certain “Go-To-Run” situations.

Many beginning programmers fail to consider these “Go-To-Run” concepts at all. Once the machinery is successfully “up and running” there’s little if any time left over to worry about “what if?” scenarios. Unfortunately the results of simply ignoring these types of issues may range from “surprising” to “embarrassing” to “expensive” and all the way to downright “dangerous” situations. Machinery may mysteriously hang up at random times. Serial numbers may become scrambled and applied to the wrong products. Parts might be intermittently shifted to incorrect conveyor stations – and then shipped to the wrong customers. Information stored in databases may become corrupted. Machining and stamping operations may get out of synch – and the machinery might be damaged as a result. Systems already in operation may suddenly initialize themselves to startup conditions for no apparent reason – or else fail to “home” and initialize properly when they should. And the list goes on.

Beside being important to programmers, maintenance technicians also need to recognize how these “Go-To-Run” concepts can affect the operation of their plant’s PLC-controlled machinery. A failure to understand the nuts-and-bolts of how the system reacts in these types of situations can make troubleshooting extremely difficult - especially since the symptoms appear to be so random and intermittent in nature. Also consider that the plant's maintenance technicians are often precisely the workers who have to deal with these types of problems – once the original programmer has finished the project and moved on to something else.

Things like "power surges" and "signal noise" often get the blame for these types of problems – and do, in fact, cause some of them. At the same time you need to realize that certain programming techniques are more susceptible to "Go-To-Run" problems than others. Once you know what to look for, many random "bugs" turn out to have definite causes which can often be tracked down and eliminated.

Throughout this series of quizzes we'll use many "counter" and "add" operations. These are meant to demonstrate, in a clear-cut manner, whether or not a certain programming method is capable of giving a "pulse" of TRUE logic. Obviously a real-world program might do much more than simply "count" such a pulse. One common example would be a "Jump to Subroutine" operation to run an "initialization" or "homing" sequence just once whenever the machinery is started up. Another might be to "Latch" (or to "Unlatch") a specific output at startup. Think of the consequences if such intended operations failed to function when required – or possibly worse - if they actually functioned when they should NOT.

Some of the questions concern the ON or OFF status of field inputs at startup time. Think of how such situations might play out in the real world. Suppose that a conveyor system stops suddenly due to a power failure – and then an operator manually moves a part along the line to the next sensor station. Later, when the power is finally restored, should the system "count" that part – or not? Will the correct serial number label be placed on that part – or not? Will that part – and all the rest after it – be sent to their proper bin locations – or will the conveyor system's "count" now be "off-by-one" until someone notices the discrepancy?

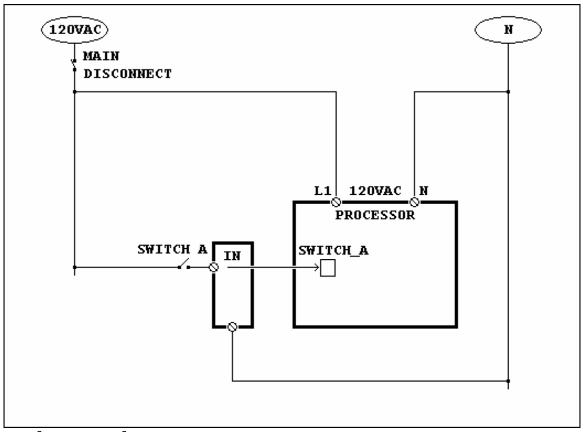
These are the types of situations that we'll be dealing with in this series of our PLC Quizzes.

Note that the "screen shot" figures used in the quiz will be taken from various RSLogix software packages. Naturally there may be some minor differences in how the rungs and the data tables are displayed from one package to another. Follow the text of the questions carefully. You should also pay careful attention to which wiring schematic is being used for each question – we'll be shifting from one to another from time to time. We'll also be making some changes from one PLC platform to another. As in real life, MicroLogix-1000, MicroLogix-1500, SLC-5/05, and PLC-5/20 systems are often mixed in together. Stay on your toes. Finally, while taking the quiz, be sure to read each question and each answer carefully – and then pick the **BEST** answer which **MOST LIKELY** answers the question.

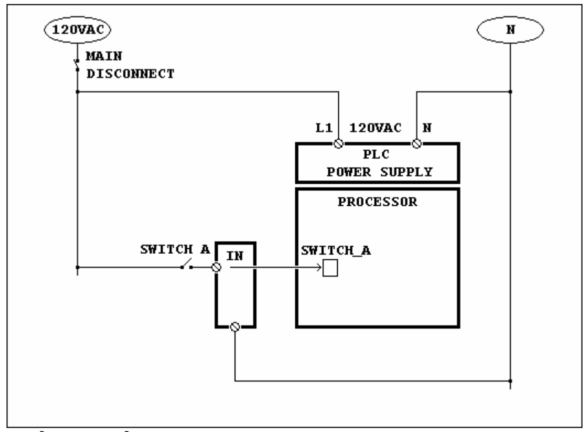
Questions 1 through 4 are just a simple "warm up" to get you started. As soon as you're finished with them, we'll go over the answers to make sure that you know how the quiz is laid out – and then we'll move on with the rest of the quiz.

Schematics for basic wiring

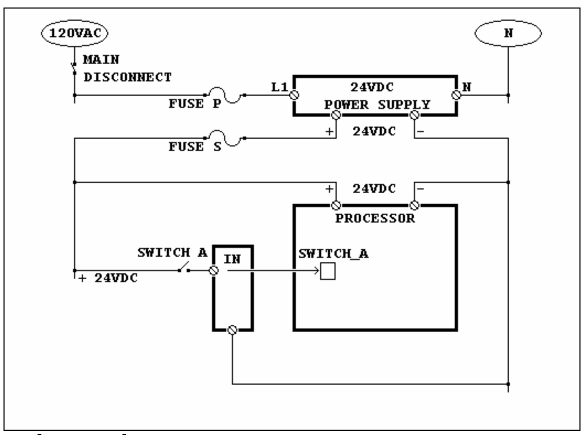
Where specific wiring schematics are mentioned in the following questions, use the drawings on this page as a guide to the wiring involved.



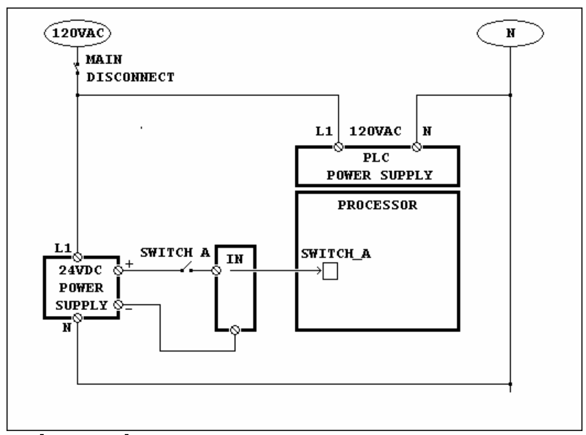
Schematic 1



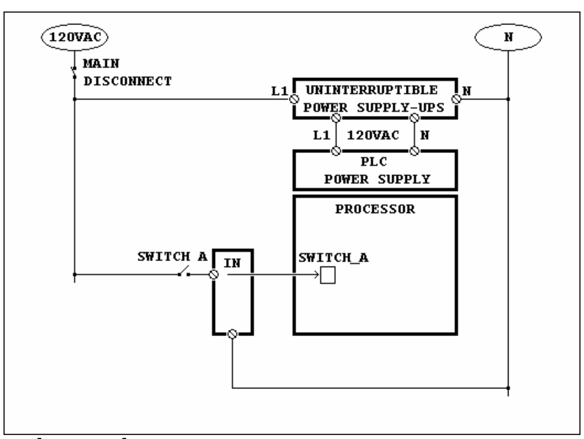
Schematic 2



Schematic 3



Schematic 4



Schematic 5

As shown in Figure A, notice that two programmers have each come up with a different approach to solving the same programming task – and notice that both approaches give identical – and perfect - results when tested under normal day-to-day conditions.

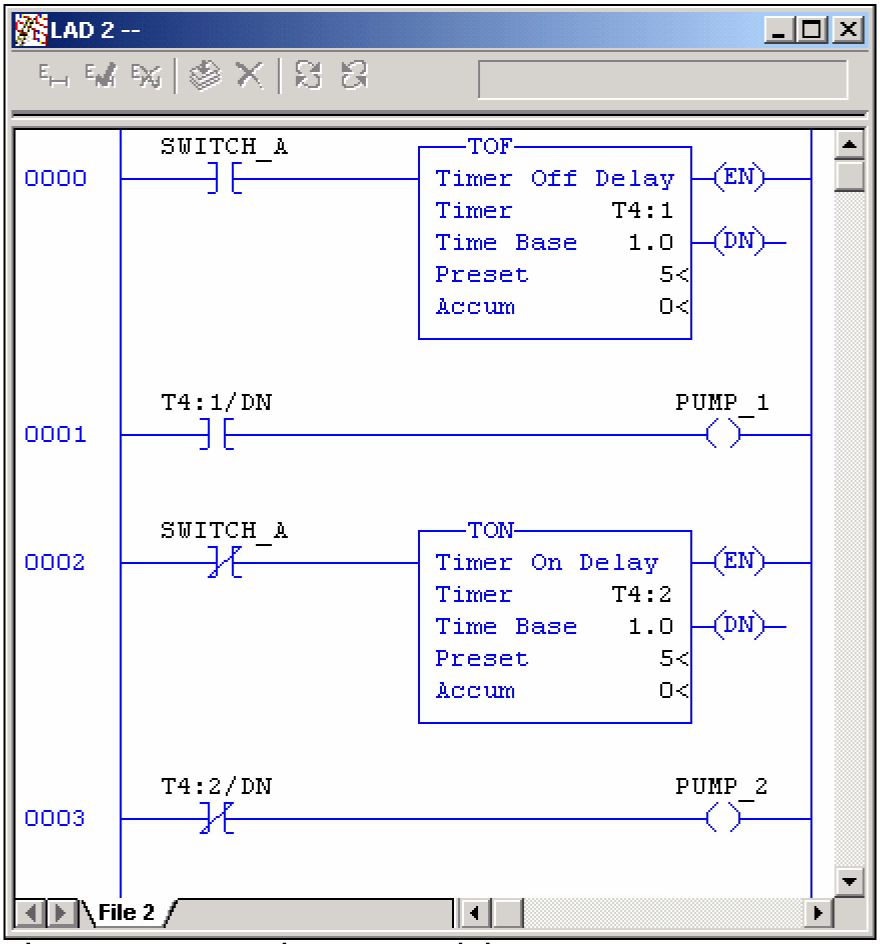


Figure A – Comparing a TOF with a TON

In Figure A, Andy has programmed Rungs 0000 and 0001 using a common TOF instruction to control PUMP_1. Bert has programmed Rungs 0002 and 0003 using his favorite "home made TOF" idea to control PUMP_2. Bert says, "I never use a TOF because the Done bit acts backwards. It's a lot simpler to just use a regular old TON instead and replace the XICs with XIOs. That way I get exactly the same results – but without the confusion of learning two separate timer instructions."

The program has been tested by repeatedly cycling SWITCH_A in the field ON and then OFF. While SWITCH_A in the field is ON, PUMP_1 and PUMP_2 both run continuously. Each time SWITCH_A in the field is turned OFF, PUMP_1 and PUMP_2 both continue to run for 5 seconds – and then they both shut down. This is exactly the desired "normal" operation for the system.

Both programmers are convinced that the two programming approaches give identical results.

Question 1 - Suppose that an MLX-1000 processor contains the program shown in Figure A. Suppose that the system is wired as shown in Schematic 1 (the output devices are not shown.) Suppose that the processor is currently in the Program Mode. Suppose that SWITCH_A in the field is OFF. Suppose that PUMP_1 in the field is OFF. Suppose that PUMP_2 in the field is OFF. Suppose that the processor is placed in the Run Mode.

How will the pumps in the field operate?

Answer A – PUMP_1 in the field will stay OFF. PUMP_2 in the field will stay OFF.

Answer B – PUMP_1 in the field will be ON for 5 seconds and then turn OFF. PUMP_2 in the field will be ON for 5 seconds and then turn OFF.

Answer C – PUMP_1 in the field will stay OFF. PUMP_2 in the field will be ON for 5 seconds and then turn OFF.

Answer D – PUMP_1 in the field will be ON for 5 seconds and then turn OFF. PUMP_2 in the field will stay OFF.

Answer E – PUMP_1 in the field will be ON and will run continuously. PUMP_2 in the field will be ON and will run continuously.

Question 2 - Suppose that an MLX-1000 processor contains the program shown in Figure A. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is OFF. Suppose that PUMP_1 in the field is OFF. Suppose that PUMP_2 in the field is OFF. Suppose that a plant-wide power failure shuts down the system for several minutes. Suppose that SWITCH_A in the field remains OFF. Suppose that the power is restored.

How will the pumps in the field operate?

Answer A – PUMP_1 in the field will stay OFF. PUMP_2 in the field will stay OFF.

Answer B – PUMP_1 in the field will be ON for 5 seconds and then turn OFF. PUMP_2 in the field will be ON for 5 seconds and then turn OFF.

Answer C – PUMP_1 in the field will stay OFF. PUMP_2 in the field will be ON for 5 seconds and then turn OFF.

Answer D – PUMP_1 in the field will be ON for 5 seconds and then turn OFF. PUMP_2 in the field will stay OFF.

Answer E – PUMP_1 in the field will be ON and will run continuously. PUMP_2 in the field will be ON and will run continuously.

Question 3 - Suppose that an MLX-1000 processor contains the program shown in Figure A. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is OFF. Suppose that PUMP_1 in the field is OFF. Suppose that PUMP_2 in the field is OFF. Suppose that the plant's power "flickers" off for about one-half second and is then quickly restored.

How will the pumps in the field operate?

Answer A – PUMP_1 in the field will stay OFF. PUMP_2 in the field will stay OFF.

Answer B – PUMP_1 in the field will be ON for 5 seconds and then turn OFF. PUMP_2 in the field will be ON for 5 seconds and then turn OFF.

Answer C – PUMP_1 in the field will stay OFF. PUMP_2 in the field will be ON for 5 seconds and then turn OFF.

Answer D – PUMP_1 in the field will be ON for 5 seconds and then turn OFF. PUMP_2 in the field will stay OFF.

Answer E – PUMP_1 in the field will be ON and will run continuously. PUMP_2 in the field will be ON and will run continuously.

Question 4 - Suppose that a PLC-5 processor contains the program shown in Figure A. Suppose that the system is wired as shown in Schematic 2. Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is OFF. Suppose that PUMP_1 in the field is OFF. Suppose that PUMP_2 in the field is OFF. Suppose that the plant's power "flickers" off for about one-half second and is then quickly restored.

How will the pumps in the field operate?

Answer A – PUMP_1 in the field will stay OFF. PUMP_2 in the field will stay OFF.

Answer B – PUMP_1 in the field will be ON for 5 seconds and then turn OFF. PUMP_2 in the field will be ON for 5 seconds and then turn OFF.

Answer C – PUMP_1 in the field will stay OFF. PUMP_2 in the field will be ON for 5 seconds and then turn OFF.

Answer D – PUMP_1 in the field will be ON for 5 seconds and then turn OFF. PUMP_2 in the field will stay OFF.

Answer E – PUMP_1 in the field will be ON and will run continuously. PUMP_2 in the field will be ON and will run continuously.

Since these first four are just "warm up" questions, we'll give you the answers before we go any further – just to show you what types of situations that we'll be dealing with in this edition of the Email PLC Quiz. The basic idea is that PLC programs often work PERFECTLY in normal day-to-day operations – but then may give surprising (and embarrassing – or even DANGEROUS) results when the system is restarted after a shutdown; or experiences a "flicker" in the plant's power supply; or is switched between the Program Mode and Run Mode; and so on. For discussion purposes, we'll categorize most of these types of scenarios as "Go-To-Run" situations – but as you'll see, that term covers a LOT of territory.

SPOILER ALERT! – THE ANSWERS FOR QUESTIONS 1 THROUGH 4 ARE SHOWN BELOW!

Answers to “Warm Up” Questions 1 through 4

Question 1 - Answer C. Notice that when going from Program Mode to Run Mode, Bert’s unorthodox programming method gave a different result from the straightforward method that Andy used. Be sure to remember that both methods worked perfectly when tested under normal day-to-day operating conditions.

Question 2 – Answer C. Here again, the method used for PUMP_2 gave a different result from the method used for PUMP_1. The big question to be asked is this: “Is it desirable for PUMP_2 to suddenly turn on after a power failure is corrected – even though the control switch hasn’t been turned on?” If the answer is “no” then this could be a serious “Go-To-Run-Gotcha” for our young friend Bert the programmer.

Question 3 – Answer A. Here the two programming methods gave identical results – but don’t miss the bigger point. Notice that in Question 2 the plant’s power failed completely and was off for several minutes. In Question 3 the power only “flickered” off and then came right back on again - and this time PUMP_2 did NOT come on and run. The point is that all “power problems” are not created equal – and the system can react differently to different types of power disturbances.

Question 4 – Answer C. Here we had exactly the same “flicker” problem as in Question 3 – but notice that we didn’t get the same results this time. That’s because this time we used a PLC-5 system instead of the MicroLogix-1000 that we started out with. In this question PUMP_2 DID come on even after the power only “flickered” off for a fraction of a second. The point is that all Allen-Bradley systems do NOT always work exactly the same way.

Remember that our quiz is titled “Surprises at Go-To-Run” – and here are some of the surprises we’ve seen so far.

(1) Two different programming methods gave identical results when tested under normal day-to-day operating conditions. But – surprise - they each gave a **different** result when the processor was switched from the Program Mode to the Run Mode.

(2) Two different programming methods gave identical results when tested under normal day-to-day operating conditions. But – surprise - they each gave a **different** result when the plant’s power failed for several minutes.

(3) The system reacted one way when the plant’s power failed for several minutes. But – surprise – the system reacted a **different** way when a brief power “flicker” occurred.

(4) The system reacted one way to a brief power “flicker” when an MLX-1000 platform was involved. But – surprise – the system reacted a **different** way when a PLC-5 platform was used.

Seem confusing? Well, it is to most programmers and technicians – even to some who have many years of experience. Think about the intermittent and apparently random problems these types of surprises can cause while debugging and troubleshooting systems in the field. And remember, we're just getting started.

Here are a few hints that might be helpful as we move forward.

(1) In each case, ask yourself whether or not the test conditions caused the processor to go through a "Pre-Scan" operation.

(2) If a Pre-Scan occurred, how did it affect each of the instructions contained in the ladder logic?

(3) If a power disturbance was involved, what effect (if any) did it have on how the PLC processor "saw" the field inputs?

(4) How does this particular PLC platform (MLX-1000, PLC-5, etc.) react to the specific test conditions?

Those are the key concepts involved. We'll have more discussion about them in the Answer section of the quiz. Just remember – this is a QUIZ to see how much you already know about the material. It's primarily meant to be entertaining and thought-provoking . Hopefully it will also turn out to be educational – but it is NOT intended to be a "lesson" in the general sense of the word.

Now that our "warm up" is out of the way, let's get on with the rest of the quiz.

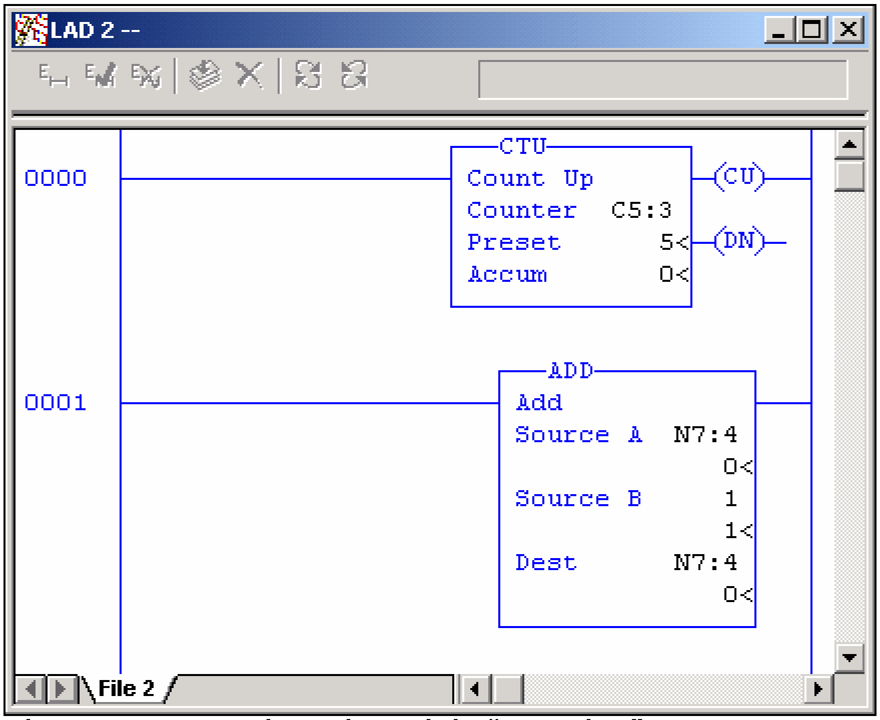


Figure B – Experimenting with "counting" scans

A team of several beginner programmers are trying to count how many times their MLX-1000 processor "scans" through their ladder logic rungs. Suppose that the system is wired as shown in Schematic 1.

Suppose that Andy has programmed Rung 0000 in Figure B using a standard CTU instruction. Andy says, "The unconditional logic of Rung 0000 is guaranteed to be TRUE, so each time the processor scans that rung, the Accumulator of C5:3 will increment by one. The counter will very rapidly go higher and higher as long as we let the system keep running."

Bert disagrees and says, "No, the logic must make a transition (a change) from FALSE to TRUE to make a CTU count up. You don't have any type of condition in your Rung 0000, so the counter will only count up ONCE – when you first go into the Run Mode. After that, the rung logic will always remain TRUE, so the Accumulator of C5:3 won't increment again. It will just go to a value of ONE – and then stay there."

Carl has another opinion and says, "Andy's right that each time the processor scans Rung 0000 the Accumulator of C5:3 will increment by one – but the counter won't go any higher than the Preset of 5. Once it gets to that value, the Done bit will come ON, and the Accumulator will stop increasing."

Dave says, "That's why I've programmed the ADD operation in Rung 0001. The ADD doesn't have a Preset to worry about. So this way, I can increment the value stored at N7:4 each time the processor scans Rung 0001. The value of N7:4 will continually increase all day long if we let it keep running."

Earl says, "Neither rung is going to do anything at all – unless you put some type of condition in each one. How will the processor know when to count or when to add unless you do?"

After some discussion, the programmers generally agree that each of the two programming approaches will accurately count the number of times that the processor scans through the rungs.

Question 5 – Suppose that an MLX-1000 processor contains the program shown in Figure B. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is currently in the Program Mode. Suppose that C5:3.ACC contains the value ZERO. Suppose that N7:4 contains the value ZERO. Suppose that the processor is placed in the Run Mode.

What will happen to the values stored at C5:3.ACC and at N7:4?

Answer A – The value of C5:3.ACC will stay at ZERO. The value of N7:4 will stay at ZERO.

Answer B - The value of C5:3.ACC will stay at ZERO. The value of N7:4 will rapidly increase to 32767 and then stop increasing when the processor faults.

Answer C - The value of C5:3.ACC will increment to ONE and then stay there. The value of N7:4 will increment to ONE and then stay there.

Answer D - The value of C5:3.ACC will increment to ONE and then stay there. The value of N7:4 will rapidly increase to 32767 and then stop increasing when the processor faults.

Answer E - The value of C5:3.ACC will rapidly increase to 32767 and then stop increasing. The value of N7:4 will rapidly increase to 32767 and then stop increasing when the processor faults.

Answer F - The value of C5:3.ACC will stay at ZERO. The value of N7:4 will rapidly increase to 32767 and then roll over into negative numbers.

Answer G - The value of C5:3.ACC will rapidly increase to the Preset of FIVE and then stop increasing. The value of N7:4 will rapidly increase to 32767 and then stop increasing when the processor faults.

Answer H - The value of C5:3.ACC will rapidly increase to 32767 and then roll over into negative numbers. The value of N7:4 will rapidly increase to 32767 and then roll over into negative numbers.

Question 6 – Suppose that a PLC-5 processor contains the same program shown in Figure B (note the change in platforms). Suppose that the system is wired as shown in Schematic 2. Suppose that the processor is currently in the Program Mode. Suppose that C5:3.ACC contains the value ZERO. Suppose that N7:4 contains the value ZERO. Suppose that the processor is placed in the Run Mode.

What will happen to the values stored at C5:3.ACC and at N7:4?

Answer A – The value of C5:3.ACC will stay at ZERO. The value of N7:4 will stay at ZERO.

Answer B - The value of C5:3.ACC will stay at ZERO. The value of N7:4 will rapidly increase to 32767 and then stop increasing when the processor faults.

Answer C - The value of C5:3.ACC will increment to ONE and then stay there. The value of N7:4 will increment to ONE and then stay there.

Answer D - The value of C5:3.ACC will increment to ONE and then stay there. The value of N7:4 will rapidly increase to 32767 and then stop increasing when the processor faults.

Answer E - The value of C5:3.ACC will rapidly increase to 32767 and then stop increasing. The value of N7:4 will rapidly increase to 32767 and then stop increasing when the processor faults.

Answer F - The value of C5:3.ACC will stay at ZERO. The value of N7:4 will rapidly increase to 32767 and then roll over into negative numbers.

Answer G - The value of C5:3.ACC will rapidly increase to the Preset of FIVE and then stop increasing. The value of N7:4 will rapidly increase to 32767 and then stop increasing when the processor faults.

Answer H - The value of C5:3.ACC will rapidly increase to 32767 and then roll over into negative numbers. The value of N7:4 will rapidly increase to 32767 and then roll over into negative numbers.

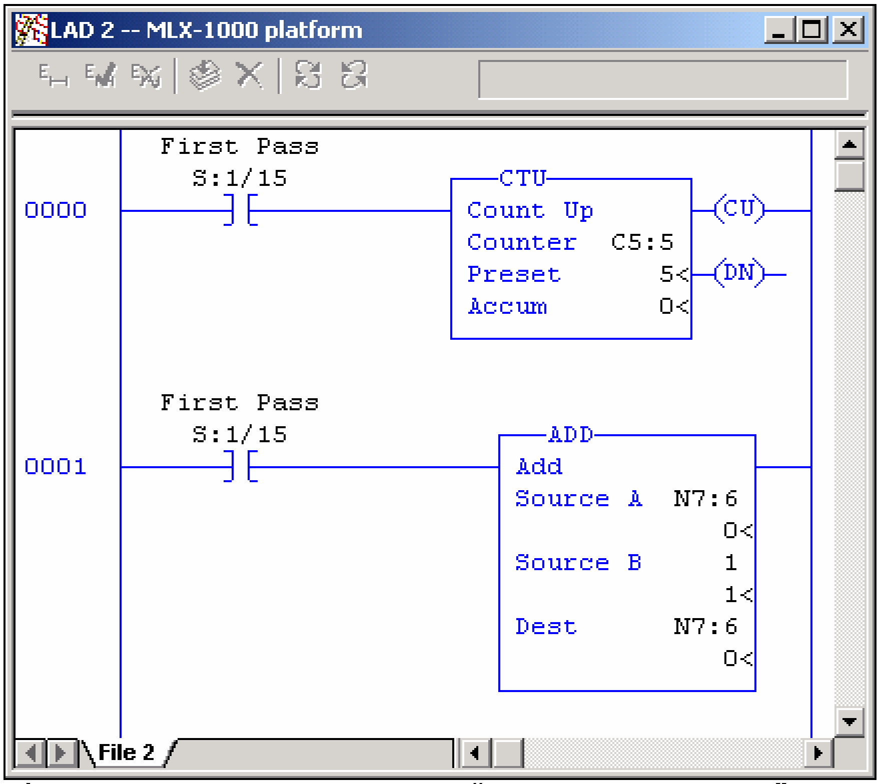


Figure C – Two ways to count "processor startups"

As shown in Figure C, Andy has just programmed Rung 0000 using a CTU instruction to track how many times his brand new processor starts up. Bert has programmed Rung 0001 using an ADD instruction to accomplish the same task.

Both programmers are convinced that the two programming approaches will properly count the start up events – and give identical results.

Question 7 – Suppose that a brand new MLX-1000 processor has just been programmed as shown in Figure C. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is still currently in the Program Mode. Suppose that C5:5.ACC contains the value ZERO. Suppose that N7:6 contains the value ZERO. Suppose that the processor is placed in the Run Mode – and then in the Program Mode – and then back in the Run Mode.

What will happen to the values stored at C5:5.ACC and at N7:6?

Answer A – C5:5.ACC will contain the value ZERO. N7:6 will contain the value ZERO.

Answer B – C5:5.ACC will contain the value ZERO. N7:6 will contain the value ONE.

Answer C – C5:5.ACC will contain the value ONE. N7:6 will contain the value ONE.

Answer D – C5:5.ACC will contain the value ZERO. N7:6 will contain the value TWO.

Answer E – C5:5.ACC will contain the value ONE. N7:6 will contain the value TWO.

Answer F – C5:5.ACC will contain the value TWO. N7:6 will contain the value TWO.

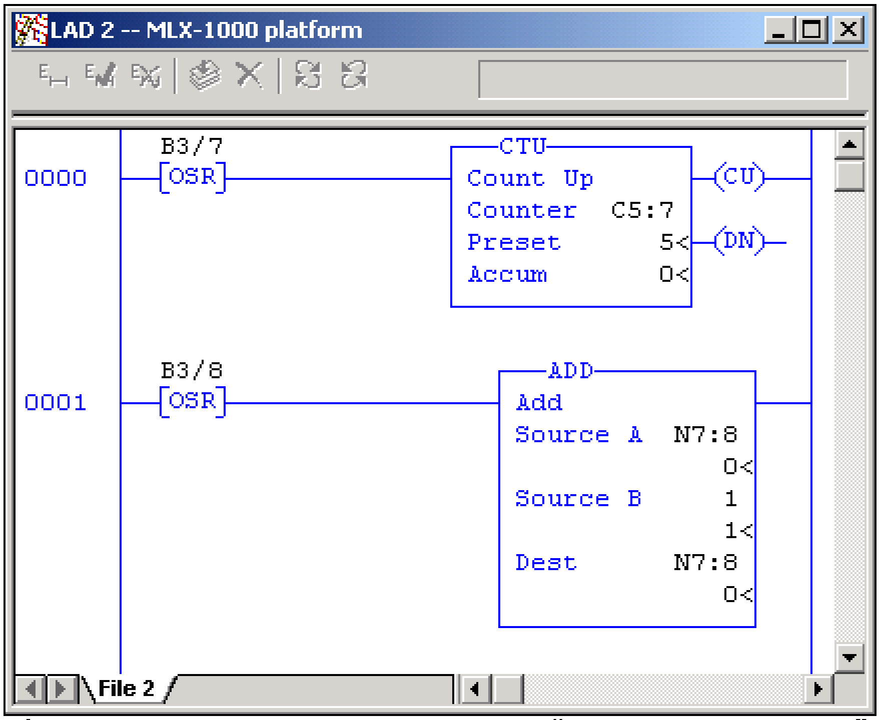


Figure D – Two more ways to count "processor startups"

As shown in Figure D, Carl has just programmed Rung 0000 using a CTU instruction to track how many times his brand new processor starts up. Dave has programmed Rung 0001 using an ADD to accomplish the same task.

Both programmers are convinced that the two programming approaches will properly count the start up events – and give identical results.

Question 8 – Suppose that a brand new MLX-1000 processor has just been programmed as shown in Figure D. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is still currently in the Program Mode. Suppose that C5:7.ACC contains the value ZERO. Suppose that N7:8 contains the value ZERO. Suppose that the processor is placed in the Run Mode – and then in the Program Mode – and then back in the Run Mode.

What will happen to the values stored at C5:7.ACC and at N7:8?

Answer A – C5:7.ACC will contain the value ZERO. N7:8 will contain the value ZERO.

Answer B – C5:7.ACC will contain the value ZERO. N7:8 will contain the value ONE.

Answer C – C5:7.ACC will contain the value ONE. N7:8 will contain the value ONE.

Answer D – C5:7.ACC will contain the value ZERO. N7:8 will contain the value TWO.

Answer E – C5:7.ACC will contain the value ONE. N7:8 will contain the value TWO.

Answer F – C5:7.ACC will contain the value TWO. N7:8 will contain the value TWO.

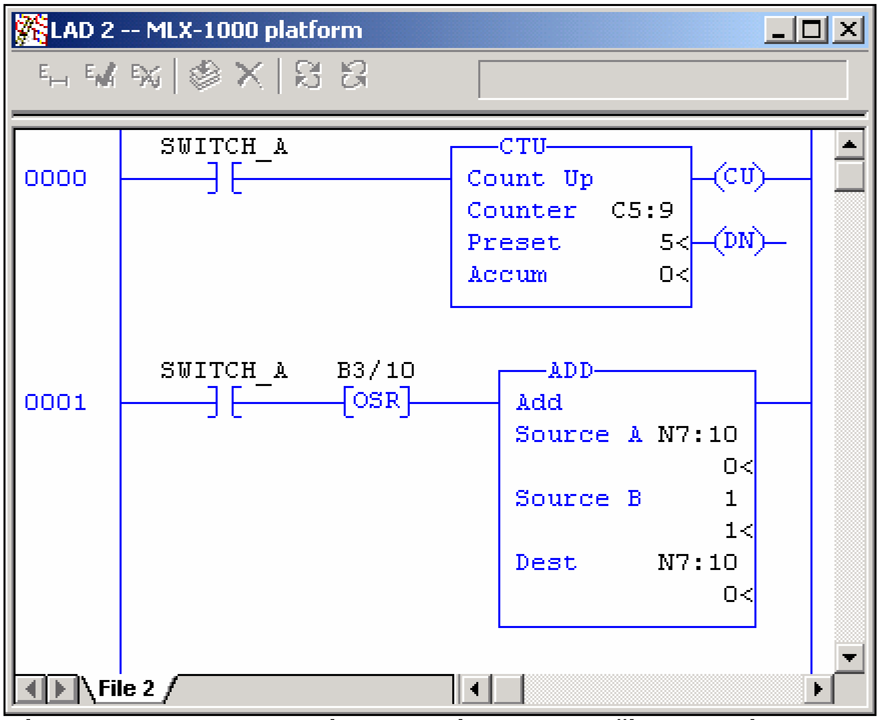


Figure E – Count Up instruction vs. a “home made counter”

In Figure E, Andy has programmed Rung 0000 using a common CTU instruction. Bert has programmed Rung 0001 using his favorite “home made counter” idea. The program has been tested by repeatedly cycling SWITCH_A in the field OFF and then ON. Each time SWITCH_A is turned ON, the value stored in C5:9.ACC increments by ONE – and the value stored at N7:10 increments by ONE.

Both programmers are convinced that the two programming approaches give identical results.

Question 9 - Suppose that an MLX-1000 processor contains the program shown in Figure E. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is ON. Suppose that C5:9.ACC contains the value ZERO. Suppose that N7:10 contains the value ZERO. Suppose that the processor is placed in the Program Mode. Suppose that SWITCH_A in the field remains ON. Suppose that the processor is placed back in the Run Mode.

What values will now be stored at C5:9.ACC and at N7:10?

Answer A - C5:9.ACC will contain the value ZERO. N7:10 will contain the value ZERO.

Answer B - C5:9.ACC will contain the value ONE. N7:10 will contain the value ZERO.

Answer C - C5:9.ACC will contain the value ZERO. N7:10 will contain the value ONE.

Answer D - C5:9.ACC will contain the value ONE. N7:10 will contain the value ONE.

Question 10 - Suppose that an MLX-1000 processor contains the program shown in Figure E. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is OFF. Suppose that C5:9.ACC contains the value ZERO. Suppose that N7:10 contains the value ZERO. Suppose that the processor is placed in the Program Mode. Suppose that SWITCH_A in the field is turned ON. Suppose that the processor is placed back in the Run Mode.

What values will now be stored at C5:9.ACC and at N7:10?

Answer A - C5:9.ACC will contain the value ZERO. N7:10 will contain the value ZERO.

Answer B - C5:9.ACC will contain the value ONE. N7:10 will contain the value ZERO.

Answer C - C5:9.ACC will contain the value ZERO. N7:10 will contain the value ONE.

Answer D - C5:9.ACC will contain the value ONE. N7:10 will contain the value ONE.

Question 11 - Suppose that an MLX-1000 processor contains the program shown in Figure E. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is ON. Suppose that C5:9.ACC contains the value ZERO. Suppose that N7:10 contains the value ZERO. Suppose that a plant-wide power failure shuts down the system for several minutes. Suppose that SWITCH_A in the field remains ON. Suppose that the power is restored.

What values will now be stored at C5:9.ACC and at N7:10?

Answer A - C5:9.ACC will contain the value ZERO. N7:10 will contain the value ZERO.

Answer B - C5:9.ACC will contain the value ONE. N7:10 will contain the value ZERO.

Answer C - C5:9.ACC will contain the value ZERO. N7:10 will contain the value ONE.

Answer D - C5:9.ACC will contain the value ONE. N7:10 will contain the value ONE.

Question 12 - Suppose that an MLX-1000 processor contains the program shown in Figure E. Suppose that the system is wired as shown in Schematic 1. Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is ON. Suppose that C5:9.ACC contains the value ZERO. Suppose that N7:10 contains the value ZERO. Suppose that the plant's power "flickers" off for about one-half second and is then quickly restored.

What values will now be stored at C5:9.ACC and at N7:10?

Answer A - C5:9.ACC will contain the value ZERO. N7:10 will contain the value ZERO.

Answer B - C5:9.ACC will contain the value ONE. N7:10 will contain the value ZERO.

Answer C - C5:9.ACC will contain the value ZERO. N7:10 will contain the value ONE.

Answer D - C5:9.ACC will contain the value ONE. N7:10 will contain the value ONE.

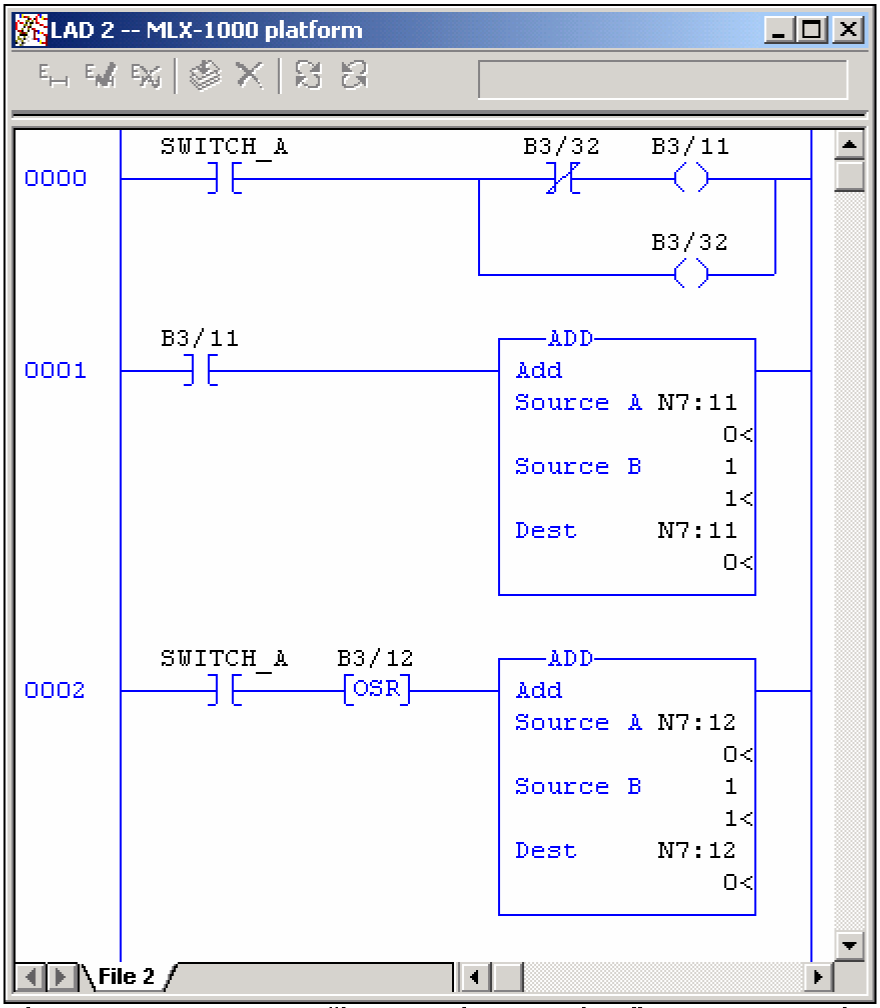


Figure F – MLX-1000 "home made one-shot" vs. an OSR instruction

In Figure F, Andy has programmed Rungs 0000 and 0001 using his favorite "home made one-shot" arrangement. Andy says, "It takes an extra rung, but I can use this method in ANY brand or type of PLC - without worrying about the differences between OSR and ONS instructions - and it works exactly the same way." Bert has programmed Rung 0002 using a common OSR instruction. The program has been tested by repeatedly cycling SWITCH_A in the field OFF and then ON. Each time SWITCH_A is turned ON, the value stored in N7:11 increments by ONE – and the value stored at N7:12 also increments by ONE.

Both programmers are convinced that the two programming approaches will give identical results – regardless of the wiring method used.

Note: Be especially careful in this last set of questions because we're going to change the way the system is electrically powered as we go. We'll keep the same program – and use an MLX-1000 platform each time – but we'll shift from AC to DC power for the last two questions. This isn't an attempt to trick you – this stuff is tricky enough as it is.

Question 13 - Suppose that an MLX-1000 processor contains the program shown in Figure F. Suppose that the system is wired as shown in Schematic 1 (note the use of AC power). Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is ON. Suppose that N7:11 contains the value ZERO. Suppose that N7:12 contains the value ZERO. Suppose that the processor is placed in the Program Mode. Suppose that SWITCH_A in the field remains ON. Suppose that the processor is placed back in the Run Mode.

What values will now be stored at N7:11 and at N7:12?

Answer A – N7:11 will contain the value ZERO. N7:12 will contain the value ZERO.

Answer B - N7:11 will contain the value ONE. N7:12 will contain the value ZERO.

Answer C - N7:11 will contain the value ZERO. N7:12 will contain the value ONE.

Answer D - N7:11 will contain the value ONE. N7:12 will contain the value ONE.

Question 14 - Suppose that an MLX-1000 processor contains the program shown in Figure F. Suppose that the system is wired as shown in Schematic 1 (again, note the use of AC power). Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is ON. Suppose that N7:11 contains the value ZERO. Suppose that N7:12 contains the value ZERO. Suppose that a plant-wide power failure shuts down the system for several minutes. Suppose that SWITCH_A in the field remains ON. Suppose that the power is restored.

What values will now be stored at N7:11 and at N7:12?

Answer A - N7:11 will contain the value ZERO. N7:12 will contain the value ZERO.

Answer B - N7:11 will contain the value ONE. N7:12 will contain the value ZERO.

Answer C - N7:11 will contain the value ZERO. N7:12 will contain the value ONE.

Answer D - N7:11 will contain the value ONE. N7:12 will contain the value ONE.

Question 15 - Suppose that an MLX-1000 processor contains the program shown in Figure F. Suppose that the system is wired as shown in Schematic 1 (again, note the use of AC power). Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is ON. Suppose that N7:11 contains the value ZERO. Suppose that N7:12 contains the value ZERO. Suppose that the plant's power "flickers" off for about one-half second and is then quickly restored.

What values will now be stored at N7:11 and at N7:12?

Answer A - N7:11 will contain the value ZERO. N7:12 will contain the value ZERO.

Answer B - N7:11 will contain the value ONE. N7:12 will contain the value ZERO.

Answer C - N7:11 will contain the value ZERO. N7:12 will contain the value ONE.

Answer D - N7:11 will contain the value ONE. N7:12 will contain the value ONE.

Question 16 - Suppose that an MLX-1000 processor contains the program shown in Figure F. Suppose that the system is wired as shown in Schematic 3 (note the change to DC power). Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is ON. Suppose that N7:11 contains the value ZERO. Suppose that N7:12 contains the value ZERO. Suppose that a plant-wide power failure shuts down the system for several minutes. Suppose that SWITCH_A in the field remains ON. Suppose that the power is restored.

What values will now be stored at N7:11 and at N7:12?

Answer A - N7:11 will contain the value ZERO. N7:12 will contain the value ZERO.

Answer B - N7:11 will contain the value ONE. N7:12 will contain the value ZERO.

Answer C - N7:11 will contain the value ZERO. N7:12 will contain the value ONE.

Answer D - N7:11 will contain the value ONE. N7:12 will contain the value ONE.

Question 17 - Suppose that an MLX-1000 processor contains the program shown in Figure F. Suppose that the system is wired as shown in Schematic 3 (again, note the use of DC power). Suppose that the processor is currently in the Run Mode. Suppose that SWITCH_A in the field is ON. Suppose that N7:11 contains the value ZERO. Suppose that N7:12 contains the value ZERO. Suppose that the plant's power "flickers" off for about one-half second and is then quickly restored.

What values will now be stored at N7:11 and at N7:12?

Answer A - N7:11 will contain the value ZERO. N7:12 will contain the value ZERO.

Answer B - N7:11 will contain the value ONE. N7:12 will contain the value ZERO.

Answer C - N7:11 will contain the value ZERO. N7:12 will contain the value ONE.

Answer D - N7:11 will contain the value ONE. N7:12 will contain the value ONE.

Summing up Beginner Level Quiz #121

As we said earlier, the material covered in this particular quiz is quite confusing to many programmers and technicians – even to some who have many years of experience. One reason for this is the fact that the concepts involved are generally not well documented in the official "books" – and what little documentation does exist is scattered here and there throughout many different publications. There just doesn't seem to be a systematic approach to defining exactly what happens when the various PLC platforms are subjected to different "Go-To-Run" situations.

Hopefully this series of our Email PLC Quizzes will at least provide a starting point of "things to look for" while analyzing the programs and systems that you work with. If you're interested in more than just simple entertainment, we recommend the following plan of attack to help you get the most educational benefit from this series of quizzes.

Carefully go over the answers and the discussion for this Quiz #121 – before moving on to the next one in the series, Quiz #122. That one is a continuation of the same types of "Go-To-Run" concepts – but at a slightly higher level. Most of the extra complications are basically due to adding in various PLC platforms to the questions. We also recommend that as you work through the quizzes, you not only write down your answers – but take the time to also write down WHY you think that a certain answer is the correct one. Comparing your notes to the actual answers will help you identify which concepts are giving you the most trouble – and allow you to concentrate your future study efforts on those specific areas.

The answers for all of our quizzes are available for downloading from our company website at www.ronbeaufort.com – look in the "Sample Lessons" section. Absolutely no registration or visitor information is ever required for access to any part of our website.

If you'd like to discuss any of the material in our PLC Quiz, just contact us and give us a chance to go over it with you. We'll be glad to answer any questions that you might have. You might also want to follow some of the "Links to PLC Resources" on our website for one of the public PLC-related internet forums where we make frequent contributions. You're welcome to start your own thread there and post your questions about any of our quizzes online. Registration is free and forum members from around the world will be glad to help you understand the concepts involved.