

Tennessee Technological University
Programmable Logic Controllers
Lab 0

1 Introduction

In this lab you gain experience with:

1. How to download project files to the PLC
2. How to download applications to the HMI
3. How to make online edits to the ladder program
4. How to make offline edits to the ladder program
5. How to toggle a bit

In this lab you will be exposed to:

1. Ladder logic
2. Logical OR and Logical AND
3. Normally closed contact (XIO)
4. Normally open contact (XIC)
5. Output coil (OTE)

1.1 How to get credit

Each lab after this will require each student to submit the completed pre-lab before they are allowed to begin working on the lab. The **pre-lab must be submitted to the TA before beginning work on the lab.** If it is not complete then you will be required to complete the pre-lab before you are allowed to begin working on the lab.

In order to get credit for completing each part of this lab, **you must personally read and complete each portion of the lab and demonstrate the completion to the TA.** Each section has one or more signature slots that must be signed by the TA to confirm that the section was completed. Each section is worth equal credit.

1.2 Twenty minute grace period

To receive full credit, the lab must be completed and demonstrated during the assigned lab time. However, if you cannot complete the lab within that time, you can complete and demonstrate the lab within the first 20 minutes of the subsequent lab time and still receive full credit. **If the lab is not completed within the assigned lab time and is not completed within the 20 minute grace period, then the lab is considered late.** If you submit the lab late, then there will be a 20% deduction compounded weekly.

1.3 Acknowledgement of author

The content of this manual is substantially the same as that in “Introduction to PLC Automation” by Dr. Jesse Robots. We are thankful for his excellent work in this area and delighted to continue to reuse his content for this course.

1.4 Safety

The PLCs and equipment in this lab use **120VAC and 24VDC**. The equipment is considered finger safe, which means that accidental contact with the electrical circuitry is *unlikely*. However, there are small gaps around the connections to the PLC and other equipment which expose small portions of bare wires or bare metal from the bussbars.

DO NOT TOUCH ANY BARE WIRES or METAL EXPOSED AROUND BUSSBARS! YOU WILL BE SHOCKED!

1.5 Lab agreement

The planning of a program is often a very social activity, however the actual writing of the code is always an individual pursuit. In this class it is very much the same. Students are welcome to verbally assist each other, but each person is required to write their own code and personally complete each lab. In this way each student will gain valuable experience with programming PLCs.

The undersigned person guarantees that any and all work demonstrated to the TA in regard to this lab is a result of their own work with no unauthorized help. Further, they acknowledge all course and safety information which has been given.

Student (Print & Sign): _____

2 Downloading to the PLC

In this section you will download the Lab 0 project to the programmable logic controller (PLC). Be careful not to **mix up downloading and uploading**. Downloading is pushing code into the PLC. Uploading is pulling the code from the PLC into the computer. In this class you should **never upload**.

The code that is in the PLC when you arrive to the lab is most likely not the code that you left in it from the previous lab. So, make sure that you save your lab progress to the lab drive or to a flash drive for safe keeping.

This section serves as **general instructions for downloading to the PLC**. Refer to this lab manual in future labs if you forget how to download to the PLC.

2.1 Get the lab files from iLearn

In this lab you will often be working with a pre-made lab PLC and HMI file. As the semester progresses this will change. However, for the next several labs you will need to logon to ilearn and download the necessary lab files to the lab PC at the beginning of lab. Download all the files in the Lab0 ilearn folder under content now.

2.2 Open Studio 5000

Studio 5000 is the development environment used to modify Allen Bradley PLC ladder logic and download to the PLC. To open the program, click the windows icon in the bottom left and navigate to a folder called **Rockwell Software**. Open the folder and select **Studio 5000**. This will launch the software.

A splash screen like the one in Fig.1 should appear. Select **Existing Project**. From the drop down that appears, select the option that says **Project File**. Navigate to the Lab PLC project that was downloaded from iLearn and select the file, then click **Open**.

Studio 5000 Splash Screen



Figure 1: Studio 5000

2.3 Set the project path

Now that Studio 5000 has been launched and the appropriate Lab project has been opened, you have to point Studio 5000 at the PLC which you will be using. This is referred to as setting the project path. **This is a very important step!**. Your classmates and TA will not be happy with you if you do not perform this step correctly, as it will cause you to **download to someone else's PLC**.

—Note 1— Accidents happen and you will be forgiven for accidentally downloading to the wrong PLC. This happens to professionals as well. However, if the TA believes that you did this intentionally, you may be asked to leave the lab.

2.3.1 What is setting the project path?

PLCs on a network are identified by their internet protocol (IP) address. The IP address for each of the PLCs in this lab corresponds with the lab seat number. Refer to Table1 to identify which PLC IP address corresponds with your seat number.

Setting the project path is what Allen Bradley calls choosing the IP address of the PLC to which you will download code.

Table 1: PLC IP addresses

Seat #	IP Address	Seat #	IP Address
1	192.168.100.201	2	192.168.100.202
3	192.168.100.203	4	192.168.100.204
5	192.168.100.205	6	192.168.100.206
7	192.168.100.207	8	192.168.100.208
9	192.168.100.209	10	192.168.100.210
11	192.168.100.211	12	192.168.100.212

2.3.2 How do you set the project path?

To set the project path, select the item in the menu bar called **Communications**. From the drop down, select **Who Active**. Here is where you will choose the PLC project path. Scroll to the PLC with the IP

address associated with your seat and click the PLC to highlight it. Then select **Set Project Path**.

—Note 2— If after highlighting the PLC, the option to set project path is still grayed out then you have either not selected the correct item or the project path is already pointing at your PLC.

There is a plus icon beside the PLC that allows you to view items associated with that particular PLC. However, to set the project path you must select the outer most object. If this is confusing, refer to Fig.2.

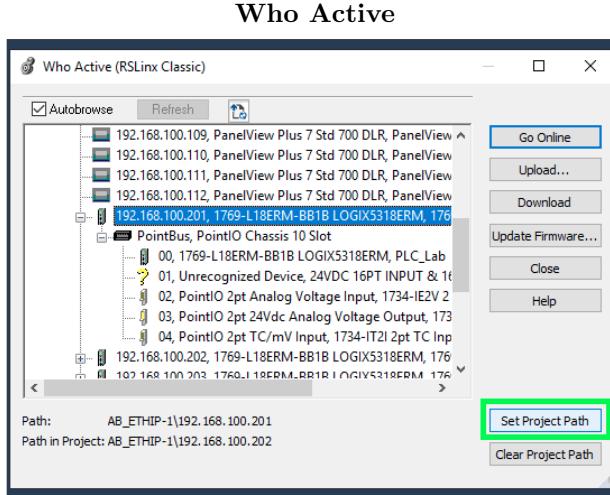


Figure 2: Who Active

—Note 3— In Fig.2 the user was sitting in seat number 1. You most likely will not select the same PLC IP address shown in the image. Refer to Table1 to verify which IP address is associated with your seat.

Do not select any other option other than Set Project Path! After setting the project path, the option will be greyed out. Close the dialog window after setting the project path.

2.4 Download to the PLC

It is now time to download to the PLC. Again click the **Communications** option in the menu. From the dropdown, select **Download**.

—Note 4— If the download option is grayed out, then you have either not set the project path or you are currently online with the PLC.

If you are online with the PLC, you will have to go to the Communications menu and select Go Offline before being able to Download.

After selecting to download the code, a dialog window will appear. Click **Download** at the bottom of the dialog. For reference see Fig.3.

—Note 5— In an industrial environment downloading can cause things to behave dangerously but in this lab environment we can ignore this warning.

—Note 6— Before downloading the code to the PLC, Studio 5000 will automatically compile what you have written. If there is a compilation error, you will be informed and the download will be aborted.

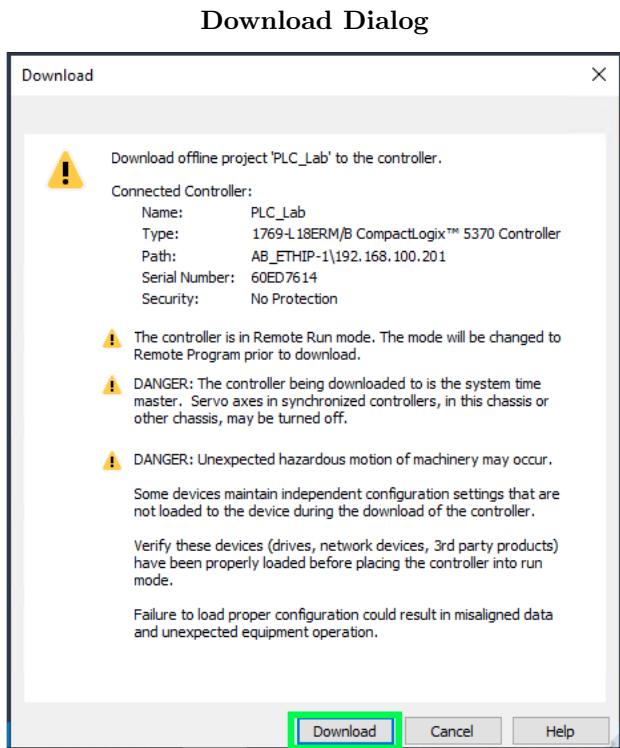


Figure 3: Download dialog

2.5 Ensure that the PLC is in run mode

After downloading the code to the PLC the code will not execute until the PLC is in run mode. Typically, after downloading to the PLC a dialog window will appear asking if you would like to return to run mode. Select, yes.

If for some reason the prompt does not appear, go to the Communications menu and select Go Online. Then go to the Communicaitons menu and select Run Mode.

2.6 Open the Main Routine

To view the code that is in the project, open the tasks dropdown in the left side menu. Here is where "routines" are kept. In each lab there will be a MainRoutine that houses the code. Refer to Fig.4. Double click the MainRoutine to open and view the code.

In Fig.5 the code housed in MainRoutine is displayed. Notice there is a comment denoting the rung which you will be editing later in the lab. Also notice that the side bars of the "ladder" are green. This means that this code is currently "being scanned" or "being executed" by the PLC. If the PLC were not in run mode then the side bars would not be green. Also, if the user were not online with the PLC the side bars would not be green.

*—Note 7— The side bars being green is meant to signify that they are energized. Ladder logic is intended to be read like an electrical schematic. Things that are green are energized. Notice that the **normally open contacts** associated with Lab0.October and Lab0.November are grey as is the **coil** associated with Lab0.Its_Fall. This means that these are not energized.*

2.7 Go Online

After downloading and ensuring that the PLC is in Run Mode, it is time to Go Online. Going online allows you to see the actual current state of the PLC and monitor the code as it executes. **Go to the**

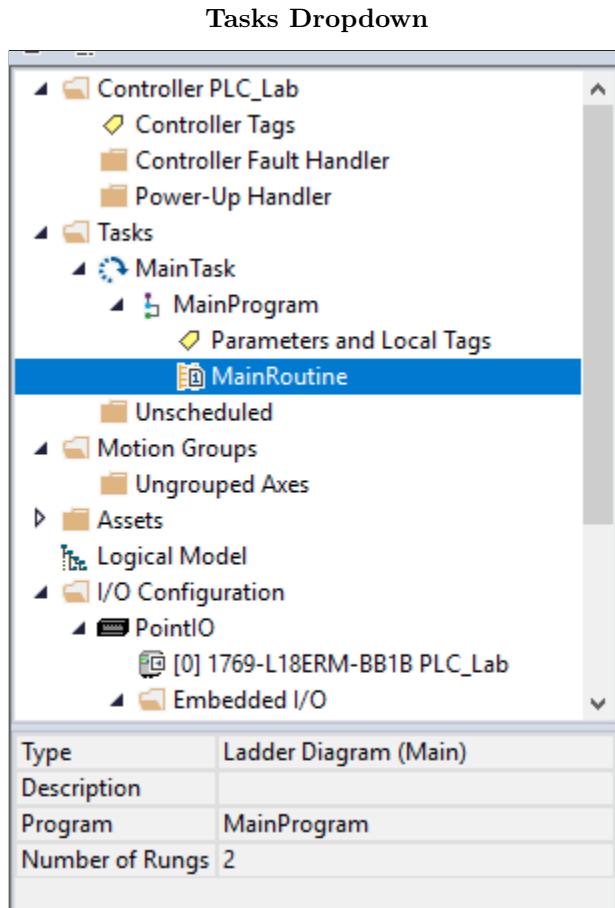


Figure 4: Tasks Dropdown and Main Routine

communications menu and select Go Online.

—Note 8— The code that is in Lab 0 is supposed to decide if it is Fall based on the month. However, you may notice that there is a logical error. We will fix this later on.

TA Signature 1: _____

3 Downloading to the HMI

PLCs work hand in hand with the human machine interface (HMI). The HMI is how operators are able to give input to and view the state of the machine. In this section you will learn how to restore an archived HMI application, generate a runtime application, and download the runtime application to the HMI.

3.1 Open FactoryTalk View Studio

The software used to edit HMI applications and download those applications to the HMI is called FactoryTalk View Studio. **Click the windows icon in the bottom left and navigate to FactoryTalk View Studio.**

The Lab0 MainRoutine

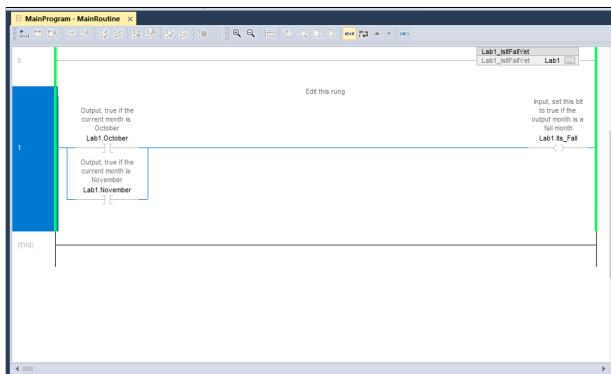


Figure 5: The code housed in the MainRoutine

When the program launches, a dialog will appear asking if you would like to open an application or create a new application. The HMI files that we distribute cannot be opened through this dialog window. **Click cancel. This will close the dialog window.**

—Note 9— To distribute HMI files they must be distributed as "backups". This is an odd idiosyncrasy of the FactoryTalk View Studio. These backups can't be opened through the initial dialog window. Rather, they must be "restored".

3.2 Restore the HMI application

To restore the HMI application that you downloaded from iLearn, go to the tools menu and select **Application Manager**. This will open the FactoryTalk View ME Application Manager. This is actually a separate program that is launched. **Highlight the button beside Restore Application and click next.**

You are then prompted for the location of the application archive. **Click the box with the three periods (ellipse) to open a file explorer window.** Navigate to the application archive file downloaded from iLearn **open the file**. This will return you to the FactoryTalk View ME Application Manager Manager. **Click next.**

You are now prompted for a name to be given to the application. **Enter Lab0 and click finish.**

—Note 10— You may be prompted to verify that you are ok with overwriting an existing HMI application with the same name. Be aware that if you have made changes to the HMI application and reopen the distributed archive file and overwrite your previous application, you may lose progress.

—Note 11— Sometimes when you attempt to restore an application and give it a name that is already used, Factory talk will not be able to overwrite the other application. In this case, give the application a name which ends with your tech username. ie. Lab0_jtroborts.

After following all the above steps, the FactoryTalk View ME Application Manager will return to the initial screen. **Close the FactoryTalk View ME Application Manager.**

3.3 Open the Restored Application

Once the application archive file has been restored, the application can be opened. **Go to the file menu and select Open Application.** In the dialog that appears, **find and highlight the application name** that you have given to the application that was restored (should be Lab0). **Click open.**

3.4 Communication Setup

The HMI needs to be able to write and read information from the PLC. For this to be possible, the HMI will need to know the IP address of the PLC.

To point the HMI at the PLC, on the left hand side menu **click the plus beside FactoryTalk linx**. Then **double click Communication Setup**. Refer to Fig.6.

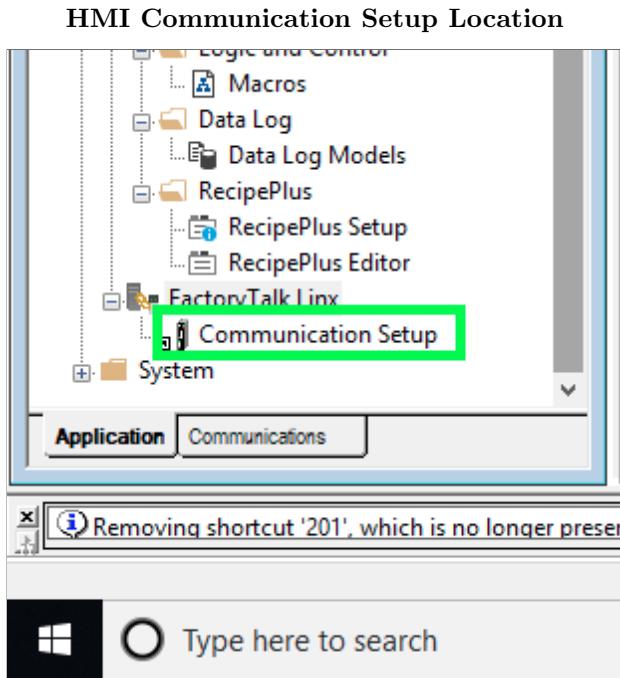


Figure 6: Where to find the HMI Communication Setup

In the window that opens, navigate to the PLC associated with your lab seat. **Click the plus beside the PLC associated with your lab seat. Click the plus beside the item named PointBus. Now, highlight the item 0.** Refer to Fig.7 for clarification.

Note 12— In Fig.7 the user was sitting in seat number 1. You most likely will not select the same PLC IP address shown in the image. Refer to Table1 to verify which IP address is associated with your seat.

With the correct PLC highlighted, **highlight PLC in the Device shortcuts to the left and click apply**. Next, **click copy from design to runtime**. Refer to Fig.8 to clarify.

The last step to setting up the communication between the PLC and HMI is to accept the communication settings that you have now put in place. To do this click OK in the lower right of the window. Refer to Fig.9.

3.5 Create the Runtime Application

Go to the Application menu and select **Create Runtime Application**. In the dialog window that appears, **select save**. This compiles the application into an HMI "runtime". A "Runtime" is what is downloadable to the HMI.

3.6 Download the HMI application

Go to the tools menu and select **Transfer Utility**. This opens another application which is used to transfer the HMI runtime application to the physical HMI. In the window that appears do the following:

Select the PLC in the HMI Software

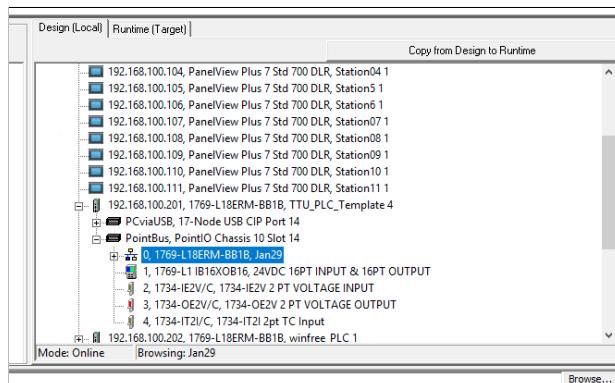


Figure 7: PLC Selection in HMI Communication Setup

Copy from design to runtime

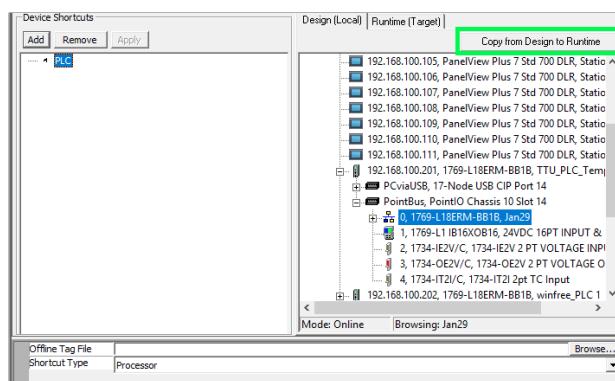


Figure 8: Location of copy from design to runtime button

1. Click the box with the ellipse next to the source file entry bar
2. Select the application you created and click open
3. Check the box beside **run application at startup**
4. Check the box beside **replace communications when transferring the HMI application**

Next you must select the IP address associated with the HMI to which you would like to download the runtime application. The HMI IP address is also associated with your lab seat. Refer to Table2 to find the IP address of the HMI associated with your lab seat.

After identifying the correct HMI IP address, **highlight the appropriate HMI in the transfer utility window and select download**.

—Note 13— You may be prompted that an application with this name already exists in the HMI. It is ok to overwrite the existing application.

3.7 Testing the HMI application

Once the download completes and the HMI reboots, the application should be running on the HMI screen. There should be a button and display.

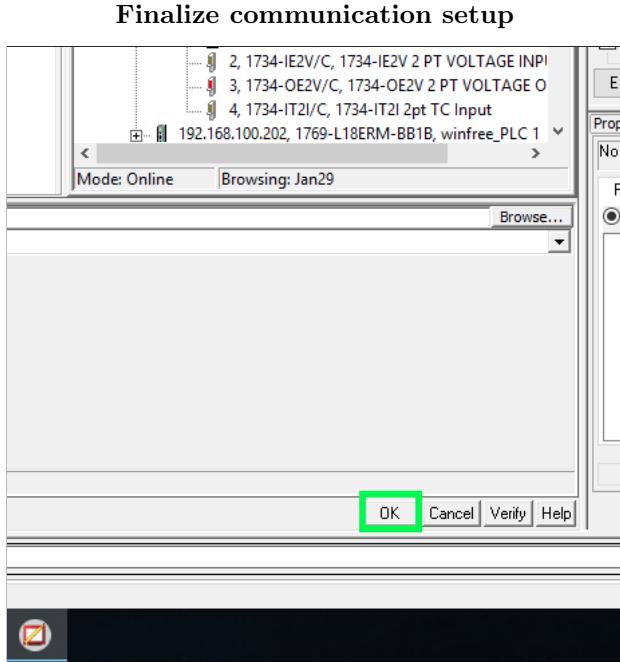


Figure 9: Ok to finalize HMI/PLC communication setup

Table 2: HMI IP addresses			
Seat #	IP Address	Seat #	IP Address
1	192.168.100.101	2	192.168.100.102
3	192.168.100.103	4	192.168.100.104
5	192.168.100.105	6	192.168.100.106
7	192.168.100.107	8	192.168.100.108
9	192.168.100.109	10	192.168.100.110
11	192.168.100.111	12	192.168.100.112

The button has the text "check code". This will check to see if the edits you make to the code in the coming sections is correct. If it is correct, the HMI will tell you that you have correctly written the code. If you hold down the button it will display the word release, instructing you to stop pressing the button.

Note 14— For the first several labs we will provide an HMI application that you must download to the HMI. The HMI application will work with code built into the lab PLC file that we provide to notify you if your code is correct and give you feedback.

TA Signature 2: _____

4 Online ladder program edits

There are two ways to edit the PLC program. The first is called an online edit and the second is an offline program edit. Online program edits allow the programmer to change the code while the PLC is still in Run mode. An offline PLC program edit requires a download after the edit is complete. Anytime a programmer

downloads to the PLC, the PLC is automatically taken out of Run mode and put into Program mode. Typically, after the download completes the programmer is prompted to put the PLC back into Run mode.

—Note 15— In a laboratory environment it is fine to stop the PLC from executing briefly to perform a download. However, imagine what would happen if the PLC that is used to control the Tennessee Tornado at DollyWood were to be taken out of Run mode in the middle of a ride!

In this section you will be making an online edit to the program to fix the logical error in the lab 0 program.

4.1 What's wrong with the logic

Before you can edit the program, you must figure out what is wrong with the logic. In Fig.5, the code is shown. The code is intended to decide if it is Fall based on the month. This doesn't mean that it is deciding if it is Fall based on the current month in the real world, but rather based on which month is "on".

The add on instruction (AOI) in the upper right hand corner of Fig.5 called Lab0_IsItFallYet has several attributes associated with it. Some are inputs and some are outputs. Particularly, it has the attributes shown in Table3.

To access any of these attributes, the dot operator is used. This is why above the contact on the left side of the rung of code in Fig.5 it says "Lab0.October". Lab0 is the name given to the **tag** that houses all the memory for the Lab0_IsItFallYet AOI.

—Note 16— An AOI (add on instruction) is an instruction that someone other than Allen Bradley has created. The AOI requires memory. The way that Allen Bradley allocates memory is in the form of tags. So, there must be a tag associated with each occurrence of an AOI in the code.

So, if the "Lab0.October" bit (boolean tag) is true (or on, or energized) that would mean that it is October. If the "Lab0.October" bit is on then the normally open contact it is associated with will be energized. Think of this like a light switch. If the "Lab0.October" light switch is on, then electricity will pass through it.

Keeping with the electrical analogy, now that the switch (referred to as a contact) associated with "Lab0.October" is on (because "Lab0.October" is on), then the voltage on the left hand side of the ladder will pass through to the right hand side of the contact associated with "Lab0.October". This voltage will continue along the rung and be applied to the coil associated with the attribute "Lab0.Its_Fall". Energizing the a coil associated with a boolean tag sets the value of the tag to true. **So, if "Lab0.October" is on, then "Lab0.Its_Fall" is also on.**

—Note 17— The left hand side of an instruction (like the normally open contact associated with "Lab0.October") is called the rung-condition-in. The right hand side of an instruction is called the rung-condition-out. If a normally open contact is on (the boolean value associated with the contact is true), then the rung-condition-outis set equal to the rung-condition-in. However, if the normally open contact is off (the boolean value associated with the contact is false), then the rung-condition-outis always set to off (or low voltage).

What about if "Lab0.November" is on but "Lab0.October" is off? The rung-condition-outfor the "Lab0.October" contact is off. However, the rung-condition-inapplied to the "Lab0.November" is on so the rung-condition-outis also on. So, a high voltage is still applied to the "Lab0.Its_Fall" coil.

4.1.1 What does the code mean logically

This rung of code implements a logical OR. If either "Lab0.October" OR "Lab0.November" is true, then "Lab0.Its_Fall" is also true. If both "Lab0.October" AND "Lab0.November" are false, then "Lab0.Its_Fall" is false.

Table 3: Attributes available in Lab0

Attribute Name	Data Type	Type
Jan	Bool	Output
Feb	Bool	Output
March	Bool	Output
April	Bool	Output
May	Bool	Output
June	Bool	Output
July	Bool	Output
August	Bool	Output
September	Bool	Output
October	Bool	Output
November	Bool	Output
December	Bool	Output
Its_Fall	Bool	Input

4.1.2 What should it be?

In reality, October and November aren't the only Fall months. September is also a Fall month. So, we need to modify the logic so that if either "Lab0.October" OR "Lab0.November" OR "Lab0.September" is true then "Lab0.Its_Fall" is also true.

4.2 Let's fix this code

Now that we have identified the problem with the present code, it is time to fix the problem. To do so we are going to make an online edit to the program.

4.2.1 Making the rung editable

To edit a rung of code while the programmer (you) is online, **right click on the number to the left of the rung and select "Start pending rung edits"**. Two versions of the rung will then be on the screen. The top version will have a column of small letter i's to the left of the rung while the bottom has a column of small letter r's to the left of the rung.

4.2.2 Adding another OR condition

We want to add another path for the voltage to flow in the case where "Lab0.September" is true. To do this we must first modify the structure of the rung and then add the instructions.

To modify the structure of the rung, highlight the left side of the rung beside "Lab0.November". This is shown in Fig.10. The next step, shown in Fig.11, is to add a branch to the rung structure. After the branch is added, it needs to be adjusted to suit our needs. Click and drag the right side of the branch so that it matches what is shown in Fig.12.



Figure 10: Highlight the rung that is going to be structurally modified

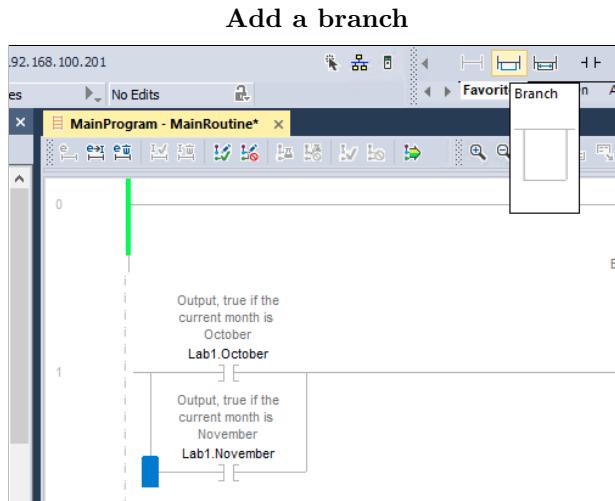


Figure 11: Add a branch to the rung structure

Click and drag the branch

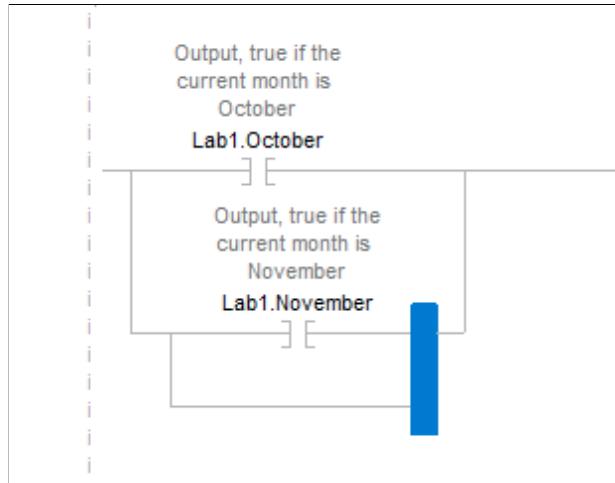


Figure 12: Click and drag the branch to suit your needs

After the rung is structurally changed to suit our purposes, you must add a contact to adjust the rung logic. Highlight the left hand side of the newly created branch and insert a normally open contact. Refer to Fig.13.

At this point there is no tag associated with the contact instruction. So, a question mark appears above the contact. To complete the changes we need to associate this new contact with "Lab0.September". To do this, select the question mark above the contact and type "Lab0.September" without the quotation marks. Press enter when it is fully typed.

4.2.3 Submitting online edits

The last step to completing our online edit is to finalize all the edits in the program. To finalize the edits click the button shown in Fig.14. A dialog window will appear asking you to confirm that you wish to finalize the edits. Select yes. If your code did not have any errors, then the code is finalized and is now running in the PLC.

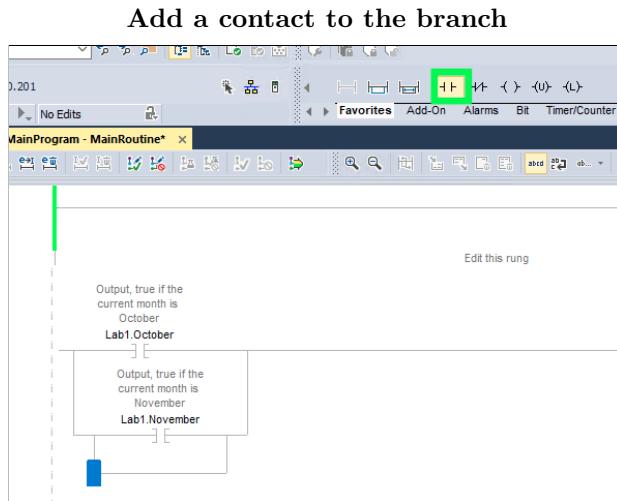


Figure 13: Add a contact to the branch

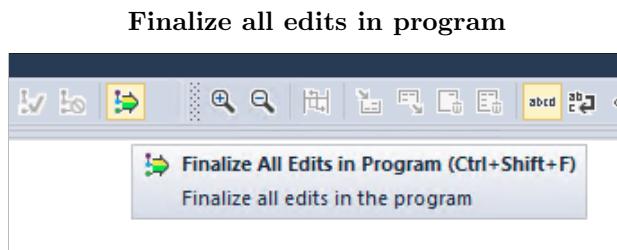


Figure 14: Finalize the online program edits

4.3 Is it Fall yet?

Now, press the check code button on the HMI. If you edited the code correctly then your HMI will say so.

TA Signature 3: _____

5 Offline ladder program edits

Another way to edit the code, as stated previously, is to take the program offline. While the code is offline, all rungs (in general) are editable. This is often convenient if you have several rungs to edit. However, you will have to perform a download after making offline changes to a program.

In this section you will make an offline edit to the program and download the edited code.

5.1 How do you know that it is Fall?

First, let's consider a logical question. How does one know that it is Fall based on the month? Certainly we have found one answer to this question. But is there another way?

The solution we have used thus far is to check if it is October OR November OR September. If any of these is true then it is also true that it is Fall.

Instead of using a logical OR we can also use the logical AND to infer the state of Fall. The logic goes like this: if it is *not* January AND *not* February AND *not* March AND *not* April AND *not* May AND *not*

June AND *not* July AND *not* August AND *not* December, then it must be Fall.

5.2 Let's code it

In order to write this into code, you will have to make use of the normally closed contact (XIO). The normally closed contact will allow the voltage to pass if the boolean tag associated with the normally closed contact is false. This allows us to code *not* January and so forth.

Structurally, the conditions won't be in branches that pass around each other. Rather, each of the normally closed contacts will be placed in series so that the only way for voltage to get to the coil associated with "Lab0.Its_Fall" is for the boolean attributes associated with all non-Fall months to be false.

—Note 18— If a normally closed contact is on (the boolean value associated with the contact is true), then the rung-condition-outis set to false. However, if the normally closed contact is off (the boolean value associated with the contact is false), then the rung-condition-outis always set equal to the rung-condition-in.

—Note 19— XIO and XIC are the actual names of the Allen Bradley normally closed and normally open contacts respectively. The acronym stands for "examine if open" and "examine if closed".

—Normally closed means that the normal (not energized) state is closed. This would be like a light switch that allowed current to flow when it was off but stopped the flow of current when the switch was in the on position (a backwards light switch). This is the same as "examine if open".

—Normally open means that the normal (not energized) state is open. This would be like a light switch that allowed current to flow when it was on but stopped the flow of current when the switch was in the off position (a typical light switch). This is the same as "examine if closed".

5.2.1 Go Offline

To set the PLC to offline, **select go offline** under the communications menu.

5.2.2 Edit the rung - Delete the conditions

Edit the rung appropriately so that it looks like the one shown in Fig.15.

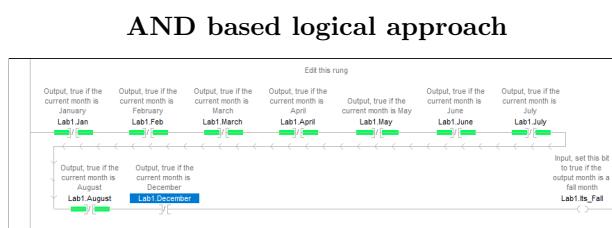


Figure 15: AND based logical approach to deciding it is Fall

5.2.3 Download

Download the code to the PLC. If you need guidance on the process, refer to section2.4.

5.2.4 Go Online

Go online. If you need guidance on the process, refer to section2.7.

5.3 Is it Fall yet?

Again press the button on the HMI. If your new logic is correct then HMI will display the same message you saw previously when you fixed the logic used to decide if it is Fall.

TA Signature 4: _____

6 Toggle a bit

Sometimes it is helpful to be able to toggle a bit from off (false) to on (true) as well as vice versa. Specifically, this is helpful when you want to test a logical circuit under certain conditions.

6.1 Toggle December

Allen Bradley provides a convenient keyboard shortcut to toggle a bit. To demonstrate this, highlight the normally closed contact associated with "Lab0.December". With it highlighted, hold the control key and press the T key (ctrl+T). This will change the boolean value of "Lab0.December".

Now, highlight the coil associated with "Lab0.Its_Fall" and use the toggle keyboard shortcut.

Why do you think that it doesn't appear to work?

—Note 20— The PLC is executing the code continuously while in Run mode. This execution happens in under a few milliseconds on an Allen Bradley PLC. So, if you attempt to toggle the state of a boolean tag which is being "driven" (another term for being associated with a coil), then the PLC quickly overwrites the result of your toggle command with the logical result from the output coil.

Demonstrate your ability to use the toggle shortcut to the TA.

TA Signature 5: _____