

Tennessee Technological University  
Programmable Logic Controllers  
Lab 2

## 1 Introduction

In this lab you gain experience with:

1. Analyzing how a multi-rung ladder logic program will execute
2. Using the output Latch (OTL)
3. Using the output Unlatch (OTU)
4. Storing the result of a rung in a tag
5. Accomplishing logical functionality using on a multi-scan approach

### 1.1 Lab Files

Go to iLearn and download the PLC and HMI files for this lab to the PC. Then download the PLC project to the PLC and the HMI application to the HMI.

### 1.2 Acceptable Instructions

You may have previous experience with PLCs and that is great! However, you are only allowed to use the instructions that we have covered thus far in the lab. So, if you have experience already, consider it a challenge to restrict yourself to only use the instructions that have been covered thus far in lecture to solve the problem!

### 1.3 How to Interface with the PLC and HMI

If you are unclear on any of the following, refer to the Lab 1 Manual:

1. Download to the PLC
2. Go online with the PLC
3. Put the PLC into Run Mode
4. Make online and offline edits to the PLC program
5. Download to the HMI
6. Toggle a boolean tag

## 1.4 How to get credit

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.

## 1.5 20 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.

## 1.6 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.**

Student (Print & Sign): \_\_\_\_\_

## 2 Box Transfer System

This section corresponds to the Lab2\_1 object in the Lab2 PLC file. **You will have to create tags to be able to build a bit latch sequence as demonstrated in lecture.**

Congratulations are in order. You have just landed a big contract with a certain online retailer that ships thousands of packages a day (you know the one). They have a heavy parcel moving machine that they use to transfer heavy packages from their conveyor to the truck. However, at the moment the machine must be controlled by a human operator. The online retailer has tasked you with automating the task.

### 2.1 How should the logic work?

They want you to write a ladder logic program that will be initiated by a button connected to a signal in the PLC called `Start_Transfer`. When `Start_Transfer` is pressed, the parcel mover will go vertically up until it reaches a switch called `Transfer_Raised`. Then it will move horizontally until it reaches the switch called `Transfer_Extended`. Then the transfer will lower until it reaches the `Transfer_Lowered` switch.

*—Note 1— Machines like this heavy parcel transfer system are typically pneumatically driven. Each of the motions made possible by a pneumatic cylinder. Each of the cylinders is connected to pneumatic valves which are controlled by the PLC. Pneumatic cylinder control is the most common way to implement motion in factories today!*

In order to tell the heavy parcel transfer machine to raise, extend, and lower use the `Raise_Transfer`,

`Extend_Transfer`, and `Lower_Transfer` signals respectively. The transfer movement doesn't happen immediately. Rather, you must hold the command for a full second before the action takes place. Play with the associated HMI display to get an idea of how the transfer movement happens. When you program this to take place automatically, the command to take an action should be held until the feedback signal becomes true. ie. if `Raise_Transfer` is true for more than 1000 milliseconds then `Transfer_Raised` should change to true.

Hint: Make sure that you turn off `Raise_Transfer` before turning on `Lower_Transfer` or nothing will happen. This is also be true with `Extend_Transfer` and `Retract_Transfer`, but there is no need in this lab to use the `Retract_Transfer` signal.

Hint: You will need to create bits to make a bit latch sequence like the one discussed in the lecture.

*—Note 2— If you energize both sides of a double-acting cylinder (a pneumatic cylinder that can be extended and retracted) with air, it will remain stationary.*

## 2.2 The Inputs and Outputs

Once you have programmed the bit latch sequence to control the parcel machine, you should be able to press the start transfer button on the HMI and see the parcel move to the truck.

To access any of the signals listed in Table1, use the syntax `Lab2_1.` followed by the attribute name.

Table 1: Attributes available in Lab2_1		
Attribute Name	Data Type	Type
<code>Start_Transfer</code>	Bool	Output
<code>Transfer_Raised</code>	Bool	Output
<code>Transfer_Lowered</code>	Bool	Output
<code>Transfer_Extended</code>	Bool	Output
<code>Transfer_Rettracted</code>	Bool	Output
<code>Raise_Transfer</code>	Bool	Input
<code>Lower_Transfer</code>	Bool	Input
<code>Extend_Transfer</code>	Bool	Input
<code>Retract_Transfer</code>	Bool	Input

Write the appropriate logic in the associated rung in the PLC file.

TA Signature 1: \_\_\_\_\_

## 3 Challenge - Toggle

This section corresponds to the `Lab2_2` object in the Lab2 PLC file.

Another weekly challenge from the competition for which you signed up! Write the necessary ladder logic to toggle the boolean tag `Toggle_Me` anytime the boolean tag `Change` has a rising edge.

You are only allowed to use the instructions that we have dealt with in the lectures up to this points (**No oneshots**). You are allowed to create extra boolean tags to help implement this functionality. If you are uncertain how to create a new boolean tag, refer to section4.

### 3.1 How should the logic work?

If `Toggle_Me` is true and `Change` goes from false to true then `Toggle_Me` should be set to false. If `Toggle_Me` is false and `Change` goes from false to true then `Toggle_Me` should be set to true.

*—Note 3— Toggle functionality is a very common requirement! Many buttons have toggle functionality and this function is typically coded in the PLC with the HMI button acting as the change state input.*

### 3.2 The Inputs and Outputs

To access any of the signals listed in Table2, use the syntax `Lab2_2.` followed by the attribute name.

Table 2: Attributes available in `Lab2_2`

Attribute Name	Data Type	Type
<code>Change</code>	Bool	Output
<code>Toggle_Me</code>	Bool	Input

Write the appropriate logic in the associated rung in the PLC file.

TA Signature 2: \_\_\_\_\_

Open Parameters and Local Tags

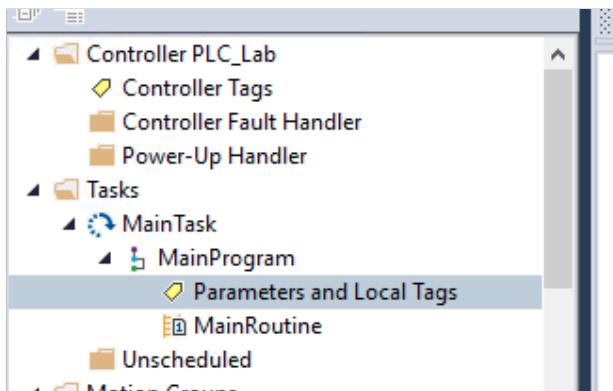


Figure 1: First step to creating a new boolean tag

## 4 How to create a boolean tag

To create another boolean tag to store the result of a boolean operation. Go to the left hand controller organizer menu. Under Main Program, double the item named Parameters and Local Tags. Refer to Fig.1.

Next, in the window that appears insure that you are on the edit tab of the parameters and tags window. In Fig.2 you can see that the edit tab is in a green box for visibility at the bottom left.

Finally, in the bottom entry in the list of tags enter the details for the tag that you are creating. The only two items that you should enter are the name and the datatype. The name must be a name that is not already taken and the datatype must be BOOL.

### Create new Boolean Tag

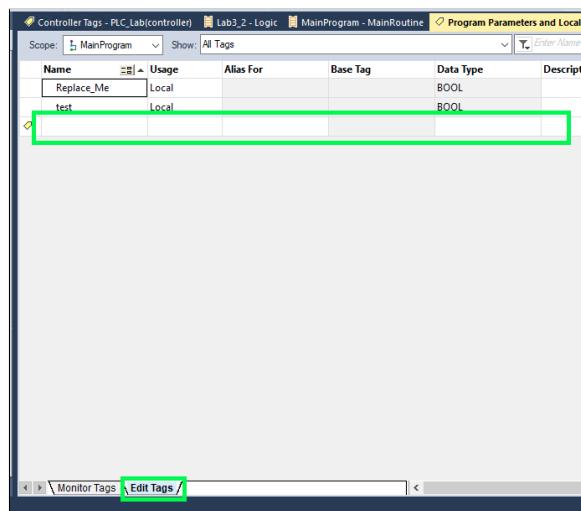


Figure 2: Second and Third step to creating a new boolean tag