L12 - Studio 5000 and Logix: Basics Lab

For Classroom Use Only!
This documentation, whether, illustrative, printed, “online” or electronic (hereinafter “Documentation”) is intended for use only as a learning aid when using Rockwell Automation approved demonstration hardware, software and firmware. The Documentation should only be used as a learning tool by qualified professionals.

The variety of uses for the hardware, software and firmware (hereinafter “Products”) described in this Documentation, mandates that those responsible for the application and use of those Products must satisfy themselves that all necessary steps have been taken to ensure that each application and actual use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards in addition to any applicable technical documents.

In no event will Rockwell Automation, Inc., or any of its affiliate or subsidiary companies (hereinafter “Rockwell Automation”) be responsible or liable for any indirect or consequential damages resulting from the use or application of the Products described in this Documentation. Rockwell Automation does not assume responsibility or liability for damages of any kind based on the alleged use of, or reliance on, this Documentation.

No patent liability is assumed by Rockwell Automation with respect to use of information, circuits, equipment, or software described in the Documentation.

Except as specifically agreed in writing as part of a maintenance or support contract, equipment users are responsible for:
- properly using, calibrating, operating, monitoring and maintaining all Products consistent with all Rockwell Automation or third-party provided instructions, warnings, recommendations and documentation;
- ensuring that only properly trained personnel use, operate and maintain the Products at all times;
- staying informed of all Product updates and alerts and implementing all updates and fixes; and
- all other factors affecting the Products that are outside of the direct control of Rockwell Automation.

Reproduction of the contents of the Documentation, in whole or in part, without written permission of Rockwell Automation is prohibited.

Throughout this manual we use the following notes to make you aware of safety considerations:

**WARNING** Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

**IMPORTANT** Identifies information that is critical for successful application and understanding of the product.

**ATTENTION** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you:
- identify a hazard
- avoid a hazard
- recognize the consequence

**SHOCK HAZARD** Labels may be located on or inside the drive to alert people that dangerous voltage may be present.

**BURN HAZARD** Labels may be located on or inside the drive to alert people that surfaces may be dangerous temperatures.
Studio 5000 Logix Designer®: Basics Lab

Contents

Before you begin ........................................................................................................................................... 4
Lab 1: Creating a Project ........................................................................................................................................... 5
Launching Studio 5000 Configuration Software ........................................................................................................ 5
Creating a New Controller Project ........................................................................................................................................... 6
Lab 2: Configuring I/O ........................................................................................................................................... 9
Adding ControlLogix I/O ........................................................................................................................................... 9
Lab 3: Monitoring/Editing Tags ........................................................................................................................................... 14
Lab 4: Adding Ladder Logic ........................................................................................................................................... 18
Adding Ladder Logic to the Main Routine ........................................................................................................ 18
Lab 5: Tasks and programs ........................................................................................................................................... 23
Lab 6: Download and Run ........................................................................................................................................... 25
Lab 7: User Defined Types (UDT) ........................................................................................................................................... 30
Creating User Defined Types ........................................................................................................................................... 30
Monitoring UDT Tags ........................................................................................................................................... 30
Lab 8: AOIs ........................................................................................................................................... 32
Using an AOI ........................................................................................................................................... 32
Lab 9: Using Logical Organizer ........................................................................................................................................... 34
Using the Logical Organizer ........................................................................................................................................... 34
Lab 10: Trends ........................................................................................................................................... 37
Running a Trend ........................................................................................................................................... 37
Before you begin

About this lab
This session provides you with an opportunity to explore the Studio 5000 Logix Designer software and the ControlLogix hardware platform. The session steps through creating a new project, programming, and downloading to a controller. This lab takes approximately 105 minutes to complete.

Tools & prerequisites
- Studio 5000 Logix Designer
- RSLinx Classic
- C:\Lab Files\Basics Logix\Logix_Basics_Lab_Demo_Project.ACD
- ControlLogix L85E demo box
Lab 1: Creating a Project

This lab section should take roughly 20 minutes to complete.

Launching Studio 5000 Configuration Software

In this section of the lab, you will launch the Studio 5000 software, which will allow you to configure and program a controller.

1. Double-click on the **Studio 5000 icon** on the Desktop to launch Studio 5000 software

The Studio 5000 Splash Screen appears

**Tip** - To see what versions of Studio 5000 you have installed on your computer, select **About** under the **Explore** section.
Creating a New Controller Project

In this portion of the lab, you will create an offline project using a ControlLogix controller.

2. Select **New Project** under the Create section.

3. When the **New Project** pop-up is displayed, select **Logix** and type ‘1756-L85E’ in the Search field.

4. Type ‘**Intro_Lab_Control_Project**’ into the name field.

5. Press the **Next** button.

6. When the **Project Configuration** window appears, **fill it in as follows**: 
Select V31

Select 1756-A7 7-Slot ControlLogix Chassis

Select Slot 0

Select No Protection

Add a project description ‘Logix Basics Lab’

Click Finish
End of lab
Lab2: Configuring I/O

We will now configure I/O for the project. To communicate with I/O modules you must add modules to the I/O Configuration folder (also referred to as the I/O tree).

Adding ControlLogix I/O

You will continue to use the project already open.

1. In the I/O Configuration Folder, right click on 1756 Backplane and select New Module.

2. The Select Module Type window appears. Type “IB16” in the search box.

3. Select the 1756-IB16IF module and click Create
Module Configuration Wizard

Whenever you add an I/O module to the system you will go through the Module Configuration Wizard. The Wizard allows you to step through the entire configuration needed for a module. You can access this information later by double clicking on a module in the I/O Configuration folder or through the tag monitor/editor.

With the Logix family, there are no dip switches or jumpers needed to configure I/O modules. I/O modules are software configured. This saves time when setting up a system. The configuration for all modules is part of the controller’s program and is downloaded to the module from the controller. This allows for ease of installation or replacement if an I/O module fails.

4. The new module window appears. Enter the Name Digital Input and Slot 3 parameters as shown below.
5. Click on the **Connection** tab to observe the **Requested Packet Interval** data. We will leave the default at **20 milliseconds**.

6. Click on **OK** to close the wizard.

7. In the **Select Module Type** window, type in “**IF8**” into the filter box and select the **1756-IF8I** module.
8. Click on *Create*. The new module window appears.

9. Fill in the name “Analog Input” and select Slot 5.

10. Select the *Configuration* tab to configure the analog module.
11. Select the Voltage radio button, and select the Input Range: 0V to 10V.

12. Click OK.

13. Close the Select Module Type dialogue.

End of lab
Lab 3: Monitoring/Editing Tags

In this lab, we will explore the Tag Monitor/Editor in Studio 5000.

1. From the Controller Organizer double-click on Controller Tags.

Notice that there are several “Local” tags. These are the tags that were created when local I/O modules were added to the I/O Config. The tag Monitor/Editor window appears. Notice in the lower left corner of the window two tabs labeled Monitor Tags and Edit Tags as shown below.

2. Expand and explore the tags for the I/O modules by clicking the arrow icon.
   - :C - Configuration tags hold the module configuration and are designated by a “:C” in the tag name.
   - :I - Input tags have “:I” in the tag name.
   - :O - Output tags have a “:O” in the name.

3. Notice the different tags types and members.
4. Click on the tab *Edit Tags*.

5. Scroll to the bottom and add three **BOOLEAN** tags, **START**, **STOP**, and **RUNNING** as shown.

6. Right click on the **Start** tag you created and select **Edit “Start” Properties**

7. Select **Alias** as a type and notice that the **Tag Properties** window changed.
8. Click on the down arrow for **Alias For**.

9. Expand **Local:3:I** by clicking on the **arrow** and select all the way down tree to **Local:3:I.Pt[0].Data**.
Tag Properties will now appear as follows:

![Tag Properties](image)

START will now be aliased to Local:3:I.Pt[0].Data, which is the first input point on the 1756-IB16IF module. Notice in the edit tags windows the alias and alias for columns now have the address selected. Using the tag START will be equivalent to using the I/O tag “Local:3:I.Pt[0].data”

![Edit Tags](image)

End of lab
Lab 4: Adding Ladder Logic

Adding Ladder Logic to the Main Routine

In this section of the lab you will add some code. During the labs we will only utilize ladder logic programming, but Logix controllers also can be programmed using Function Block, Sequential Function Charts, and Structured Text. This allows selection of the programming language that best fits an application.

1. In the Controller Organizer expand the MainProgram folder by clicking on the arrow 🔄. Once expanded, the MainProgram will appear as shown below:
2. Double-click the **MainRoutine** icon and **maximize the ladder window** if it is not maximized. This will open the routine editor. An empty rung will already exist as shown below: The red color of the rung and the circled “x” next to the rung indicate the rung is incomplete.

![Diagram of MainRoutine icon and ladder window]

3. From the instruction toolbar, left click and hold on the **Examine On (XIC)** instruction and drag the **XIC** onto **rung 0** until the **green dot** appears. Release the mouse button at the location you wish to place your instruction.

![Diagram of Examine On (XIC) instruction being dragged and a green dot appears]

Verify your rung appears like the figure below:

![Diagram of completed rung with red color and circled “x”]
4. From the instruction toolbar, left click and hold on the Output Energize (OTE) instruction onto the right of the rung.

**FYI** - If you place an instruction in the wrong location on a rung, simply click and hold on the instruction and drag it to the correct location.

Verify the rung appears as follows:

As you can see the free form editing in Studio 5000 can help speed development. You do not have to place an instruction and tie an address to it before you add the next instruction.
5. Double click on the “?” above the XIC instruction to get a drop down as shown.

![Diagram showing the drop down menu]

6. Select the **drop down** and find and select the **START** tag

![Diagram showing the selected tag]

7. The rung should look as follows

![Diagram showing the final rung layout]
8. Add **RUNNING** to the **OTE** so the rung appears as follows

See, we can quickly and easily add instructions and tags using Logix Designer!

**End of lab**
Lab 5: Tasks and programs

1. Right click on **Tasks** and select **New Task**

![New Task dialog box](image)

2. Create a *new task* of type **periodic**, name **My Task**, with a period of **250 ms**, and **OK** to create the task.

![Task properties dialog box](image)

3. Now, let’s add a program to it. Right click on the **My Task** folder and select **New Program**.

![Program creation dialog box](image)
4. Call the program **My_Program** and schedule in **My_Task** and click **OK**

5. Right click on **My_Program** and **Add a new Routine**.

6. Give the routine the name **My_Routine** and click **OK**.

That's all! We have now created a new task, program, and routine that will execute every 250 ms. We could now add any logic we wished.

**End of lab**
Lab 6: Download and Run

The layout of the L85E demo box we will be using.

1. **Close** the currently open project and save it as any name you like.  
   We will not need this project again. We will now use a previously built project.
2. Open the file

C:\Lab Files\Basics Logix\Logix_Basics_Lab_Demo_Project.ACD

Notice that the project has three tasks.

3. Select Who Active as shown
4. Select the L85E in the who window that shows up and select **Download**

![Screenshot of the Who window with L85E selected and Download button highlighted.]

5. Put the controller into **run mode** if prompted, OR, put the controller in **run mode from the Rem Prog tab**.

![Screenshot of the Rem Prog tab with Run Mode selected.]

Verify the controller is in Remote Run mode as shown below.

![Screenshot showing the controller in Remote Run mode.]

.
6. Open the MainRoutine shown below and navigate to rung 4

7. Push and hold the DI0 button on the demo box and verify the rung looks like the following

Notice the DO0 light is also on. It is wired to Local:4:O.Pt[0].Data

8. Open the Demo_Event_Routine.

9. Try any of the push buttons. Try push button DI2. Notice how the rungs behave..

You should see at least one of the ADD instructions increment when a button is pushed.

10. Right click on the Demo_Event_Task and select properties, then click configure tab.

   Notice how the event task has been configured.
11. Open the local tags of the Demo_Event_Program. Notice that the tag values can be viewed here too. Try pushing some buttons.

12. Open the Periodic Routine. Notice the Add instruction incrementing once a second.

13. Try viewing the properties of the periodic task and verify the period is 1 second.

14. Try pushing the DI3 push button several times. Notice how the DO3 light responds. It is wired to Local:4:O.Pt[3].Data

End of lab
Lab 7: User Defined Types (UDT)

Creating User Defined Types

In this section of the lab you will create a custom User Defined Type (UDT).

What is a UDT and what is it good for?
A UDT is good for organizing related data into a single structure. A UDT allows a single tag to hold multiple members. Each member can be given a unique name to describe the data it holds. The members are accessed by the main tag name, followed by a period, followed by the member name.

Continue to use the project already open.

1. Open the continuous task MainRoutine
2. Go to rung 3

Noted there is a tag based on a UDT already filled in. This tag has a member of gallons and a member of Liters.

Notice that the values of the tags are shown on the instruction. The multiply instruction converts the number in gallons to liters.

3. Click on the number just below gallons, type “34” or any desired value, and press enter. Notice that the Liters value updates automatically.

Monitoring UDT Tags

4. Double click the Parameters and Local Tags under the MainProgram and expand the Gallons_to_Liters tag. Notice the values are also shown here. Make sure to select the Monitor Tags tab.

5. The values for gallons can be modified directly in the monitor screen by changing the value in the Value column. Change the gallons value and watch that liters updates to corresponding value.
The UDT allows associated data to be stored under a single main tag instead of using completely separate tags. This makes it easier to keep track of data and keep it more organized. The UDT name itself can document what the data is for.

End of lab
Lab 8: AOIs

Using an AOI

1. Open the MainRoutine

2. Navigate to rung 6

3. Turn the DI4 dial on and off. When the dial is on, notice that the Signal_Output tag alternates between 0 and 1 cause an output light to go on and off.
4. Right click on the **Signal_Light instruction** and select **open logic**.

5. Turn the dial DI4 again on and off. Watch the inner workings of the instruction in real time!

6. Close the **Signal_Light Logic**.

7. Right click on the instruction again and choose **Open Definition**

8. Feel free to explore the tabs, especially the **Parameters** and also **Local Tags**

AOI’s allow code to be encapsulated into a single instruction. This allows common code and functionality to be clearly defined and easily reused. The AOI can be reused as many times as desired. Each AOI should typically have a unique backing tag.

End of lab
Lab 9: Using Logical Organizer

Using the Logical Organizer

1. Click on the *Logical Organizer* tab

   By default, all of the programs in the Controller Organizer are shown as an ungrouped list. Only the programs, and not the tasks, are shown. We can use the organizer to group the programs. We can use a “folder” object to group the programs.

2. To add a new folder, right click on the *Logical Model folder* -> *Add* -> *New Folder*

3. Enter the name ‘*My_Demo_Box*’ and click *OK.*
4. Click and drag the three programs onto My_Demo_Box so that they are grouped under the My_Demo_Box folder as shown.

5. Click on the Controller Organizer and notice that the programs and tasks haven't changed.
Notice that the programs we are using to run the demo box are now grouped together. Anyone looking at the Logical Organizer will have a better idea that both programs are being used to run the demo box. In general, the organizer is used to group the program code to model the physical or logical application.

End of lab
Lab 10: Trends

Trending
Basic Trending in Studio 5000 allows you to view data sampled over a time period in a graphical display. Data is sampled at a periodic rate that is configurable from 1 millisecond to 30 minutes. Studio 5000 will allow you to create a trend and save it as part of your project file.

Basic Trending has these constraints: you can trend data elements of type BOOL, SINT, INT, DINT, and REAL, you are limited to sampling eight unique data elements in a single trend.

Running a Trend
1. From the Controller Organizer, expand Trends and select Demo_Trend_Sine_Wave.
2. Start the trend by clicking on the RUN button located toward the upper left of the Trend dialog box.
3. This trend has been set up to capture sine wave data. Watch as the sine wave builds.
4. After you are satisfied it really is a sine wave, close the trend.

5. Open the trend Demo_Trend_Start_Stop

6. Run the trend

   Notice nothing happens

7. Now press the DI0 button

   Notice the trend start to run. This trend has been configured to start on the press of DI1.


   Notice there are some samples after the press of the button. The amount of samples stored before or after a press is configurable!

9. Stop the trend

10. Right click on the graph background and select “Chart Properties”.

11. Navigate to the Pens tab, the Start Trigger tab, and Stop Trigger tab and notice the configuration like that below.

   Feel free to explore! If you stopped the trend, try change some of the properties to see the effect they will have. Remember to apply and start the trend again!
By default, each tag will be independently scaled to its observed min/max values. If desired, the scaling options can be changed under the chart properties - Y axis tab.

There are also other options in the trend properties such as a start and stop trigger and pen colors.

12. When you are finished investigating the trend, click Stop and close the trend window.

End of lab