L14 - Experience Connected Components Workbench™ Software to Develop Your Basic Standalone Machine Application

For Classroom Use Only!
Important User Information

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.
Experience Connected Components Workbench Software to Develop Your Basic Standalone Machine Application

Contents

Before you begin ........................................................................................................................................... 4
About this lab ............................................................................................................................................... 4
Tools & prerequisites .................................................................................................................................. 4
Lab setup ................................................................................................................................................... 4
Exercise #1 - Create a Connected Components Workbench project and add a Micro850 controller .......... 5
Exercise #2 - Add a PowerFlex 4M drive to your project ............................................................................ 11
Exercise #3 - Program a motor start/stop rung in ladder logic .................................................................. 20
Exercise #4 - Build, download and test your Micro850 Project ................................................................. 30
Exercise #5 - Use Run Mode Change to add a structured text program to control motor speed ............... 36
Exercise #6 - Import a project, add a graphic terminal and define tags using direct referencing ............... 44
Exercise #7 - Create a graphic terminal screen, download and test .......................................................... 55
Exercise #8 - Configure VNC (Virtual Network Computing) .................................................................... 72
Optional Explore PanelView 800 Advanced Features .............................................................................. 78
Before you begin

The following should have already been verified with your demo kit by the lab instructor prior to the lab:

1. One USB cable connected between PC and Micro850 controller and a second USB cable connected between PC and 1203-USB mounted to the left of the Kinetix 3 drive.
2. Ethernet cable connected directly between the PC and the PanelView 800 terminal.
3. Black cable from back of 1203-USB with RJ-45 connector on the end plugged into the PowerFlex 4M RJ-45 port.
4. Micro850 controller firmware at v10.011 (or higher) with static IP address configured for 192.168.1.3.
5. PanelView 800 terminal firmware at v4.013 (or higher) with static IP address configured for 192.168.1.2.

About this lab

Connected Components Workbench (CCW) is the integrated design environment software package that is used to program, design, and configure your Rockwell Automation Connected Components devices such as Micro800 programmable logic controllers, PowerFlex variable frequency drives, Kinetix 3 servo drives, SMC soft-starters, Guardmaster software configurable safety relays and PanelView 800 operator interface terminals.

This lab will guide you in configuring a PowerFlex 4M drive, programming a Micro850 controller, and creating and downloading a PanelView 800 terminal application, all using Connected Components Workbench software.

Tools & prerequisites

- Software: Connected Components Workbench (CCW) Version 10.01 software – Developer Edition
- Hardware: Connected Components demo kit – DEMO-CCMICRO1

Lab setup
Exercise #1 - Create a Connected Components Workbench project and add a Micro850 controller

In this section, you will create a new Connected Components Workbench project and learn how to:

- Add a Micro850 controller to your project by selecting a new device from the 'Catalog'.
- Add and configure plug-in modules for the Micro850 controller.
1. Start the Connected Components Workbench (CCW) software by clicking the icon at the bottom left hand side of the screen.

2. Add a Micro850 to your project.

Click on New... from the Start Page. Accept the New Project defaults by clicking on Create. Expand the Controllers folder under Catalog. Expand the Micro850 folder, click on catalog number 2080-LC50-24QBB, click Select then click Add To Project. This will add a Micro850 controller to your project.
3. Notice that the Micro850 shows up both in the Project Organizer on the left-hand side as well as a large graphic in the middle.

4. Configure the Ethernet port as shown below.
5. Next, the plug-in modules used by this lab need to be added to the controller and configured. This Micro850 controller has three plug-in slots available. Plug-ins can be added to provide analog inputs and/or outputs, additional digital inputs and/or outputs, additional communication ports and other specialty functions.

6. Right click on the middle plug-in slot on the large Micro850 graphic and select a **2080-IF4** analog input plug-in module.
7. Using the pull-down menu, change the 2080-IF4 Channel 0 Input Type from **Current** to **Voltage**.
8. Similarly, right click on the third plug-in slot, select a 2080-OF2 analog output plug-in module, and change the 2080-OF2 Channel 1 Output Type to **Voltage** and the Output State to **Enabled**.

Note that the physical Micro850 controller in the demo box also has a 2080-SERIALISOL plug-in module installed in the first plug-in slot, but because we are not going to use this serial communications plug-in for this lab, we don't need to include it in this project.
Exercise #2 - Add a PowerFlex 4M drive to your project

In this section, you will learn how to:

- Add a PowerFlex 4M drive to your project using Discover.
- Configure the drive using the Startup Wizard.
9. Now that the Micro850 controller is added and configured, the next step is to add and configure the PowerFlex 4M drive. Since the drive is already connected to communicate with the PC via the 1203-USB interface mounted in the demo box, the easiest way to add it to the project is to **Discover** it. Click on the **Start Page** tab to re-open the Start Page and click on **Discover**…

10. Expand **AB_DF1-1** in the **Connection Browser** window, click on the **DSI** icon and click **OK**.
11. CCW will automatically connect to, upload and add **PowerFlex 4M_1** to the project. Click Wizards to bring up the Wizards screen and click Start to open the PowerFlex 4M Startup Wizard.

12. Click **Next** to continue to step 2 of the Wizard.
13. Click **Reset Parameters** and **Yes** to confirm. The PowerFlex 4M drive display will flash **F048**, which is the fault that indicates that the drive parameters have been reset back to factory defaults. Click **Next** to continue to step 3 of the Wizard.
14. Enter in '0.2' for the **Motor NP FLA**, then click **Next** twice to skip step 4 and continue to step 5 of the Wizard.
15. Enter ‘5.0’ for Speed Reference, click the red stop button to clear any drive fault, then click the green start button.

16. When the Speed Reference window opens, click Yes.

17. The motor should be running now in the demo case. Click the red stop button, then choose Yes for ‘Is the direction of motor rotation correct for the application?’. Click Next twice to skip step 6 and continue to step 7 of the Wizard.
18. Select **0-10V Input** from the **Speed Reference** drop-down menu. This will allow the Micro850 to control the speed of the motor using an analog output from the 2080-OF2 plug-in that is wired to the drive’s analog input. Click **Next** to continue to step 8 of the Wizard.
19. Select **2-Wire** from the **Start Source** drop-down menu. This will allow the Micro850 to start and stop the drive using a digital output that is wired to the drive’s digital input.
20. Click **Next** twice to reach step 10 and then click **Finish**.

21. Save the drive changes in the project by selecting the **Save** icon ( ) from the toolbar.
Exercise #3 - Program a motor start/stop rung in ladder logic

In this section, you will learn how to:

- Add a ladder diagram program to the Micro850 controller.
- Rename the ladder diagram program.
- Program a ‘seal-in’ circuit to start and stop the motor in ladder logic.
22. Add a Ladder Diagram program.

Right-click **Programs** under the Micro850 in your Project Organizer, and select **Add \rightarrow New LD : Ladder Diagram**.

Micro800 controllers allow you to create multiple programs as well as use multiple types of programs (such as Structured Text or Function Block Diagram) in the same controller application.

Since we’ll be creating a Motor Circuit in this program, let’s rename it **Motor_Circuit**.
24. Right-click the **Prog1** program icon and select **Rename**.

25. Rename the program, **Motor_Circuit**.

This circuit will use the DI0 (Digital Input 0) switch on the Demo box as your Start switch, and the DI1 (Digital Input 1) switch as your Stop switch. The start motor control is wired to DO9 (Digital Output 9) so that when DO9 is turned on, the motor accelerates and runs, and when DO9 is turned off, the motor decelerates to a stop.

27. Double-click the **Motor_Circuit** program icon. A ladder diagram editor will appear in the main project workspace with one empty rung. Click on the **Toolbox** tab to the far right of the screen to open the Toolbox.

28. Toggle the **Toolbox** at the top right-hand corner of the **Toolbox** window to keep it from auto-hiding.
29. The Toolbox provides all the elements needed for programming a Ladder Diagram. The elements include rung, instruction block, branch, coil, contact, return and jump as defined below.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rung</td>
<td>Represents a group of circuit elements that lead to the activation of a coil.</td>
</tr>
<tr>
<td>LD instruction block</td>
<td>A block element can be a function block, a function, a user-defined function block, or an operator.</td>
</tr>
<tr>
<td>Branch</td>
<td>Two or more instructions in parallel.</td>
</tr>
<tr>
<td>Coil</td>
<td>Represents the assignment of outputs or internal variables. In an LD program, a coil represents an action.</td>
</tr>
<tr>
<td>Contact</td>
<td>Represents the value or function of an input or internal variable.</td>
</tr>
<tr>
<td>Return</td>
<td>Represents the conditional end of a diagram output.</td>
</tr>
<tr>
<td>Jump</td>
<td>Represents the conditional and unconditional logic in the LD program that controls the execution of diagrams.</td>
</tr>
</tbody>
</table>

30. Locate the **Direct Contact** instruction in the Toolbox pane (right-hand side), and drag-and-drop it onto the left side of the first ladder rung. The “+” icon on the rung indicates the location of a drop point for the instruction.

After inserting the Direct Contact instruction, the Variable Selector window will automatically pop-up, allowing you to select the variable or input/output point to assign to this instruction.
31. In the Variable Selector window, select the **I/O – Micro850** tab.
32. We will be assigning an embedded input point to this instruction. Embedded Input/Output (I/O) variable names start with '_IO_EM_\_' followed by DI for digital input or DO for digital output, concluding with the input or output point number, starting with 0. Select _IO_EM_DI_00 for embedded input 0 and then, in the ‘Alias’ column of _IO_EM_DI_00, type ‘Start Motor switch’ and click OK.

33. Your rung should look like the following.
34. Locate the **Direct Coil** instruction in the Toolbox, and drag-and-drop it onto the far right side of the rung, and assign it to the embedded output point _IO_EM_DO_09 with alias description ‘**Motor ON/OFF**’. Your rung should look like the following.

![Diagram 1]

35. Locate the **Reverse Contact** instruction in the Toolbox and drag-and-drop it onto your rung, just to the right of the Direct Contact and assign it to embedded input point _IO_EM_DI_01 with alias description ‘**Stop Motor switch**’. Your rung should look like the following.

![Diagram 2]

36. Locate the **Branch** instruction in the Toolbox and drag-and-drop it to the left of the Direct Contact on the far left of the rung. Your rung should look like the following.

![Diagram 3]

37. Drag-and-drop a **Direct Contact** onto the Branch that you just added, and assign it to the embedded output point _IO_EM_DO_09. Notice that output points can be assigned to contacts as well as to coils. Your rung should look like the following.

![Diagram 4]
38. Next drag-and-drop an Instruction Block to the immediate right of the Start Motor switch direct contact so that it is within the branch. The Instruction Block Selector screen opens up automatically to allow you to select which of the more than 150 instructions you would like to choose from. By default the instructions are sorted alphabetically by Name, but you may also sort them by Type or Category by clicking on those column headings. We want to delay the starting of the motor until the switch has been held on for one second with a timer instruction. Click on the Category column heading, scroll to the bottom to the Time instructions, click on TON for the Timer ON-delay instruction and click OK.
39. Specify a one second Programmed/Preset Time (PT) for the timer by clicking in the upper half of the input box to the left of PT, type T#1s in the selection box and Enter. T# indicates that this is a time value and 1s is for one second. After the input (IN) to the timer has been on for one second (Elapsed Time ET=PT), the timer output (Q) turns on.

40. It is always a good programming practice to document your program rungs so that when you or someone else looks at the rung in the future, it will be clear what the intended function of the rung is. Double click within the rung comment box just above the rung and type in “Switch DI0 turns the motor on after one second and switch DI1 turns the motor off.”
Exercise #4 - Build, download and test your Micro850 Project

In this section, you will learn how to:

• Build your application and verify that there are no errors.
• Download your application to the Micro850 controller.
• Test that it works as intended.
Before you can download a project to the controller, you must build it to verify that there are no errors with the programming.

41. Build your application by right-clicking the Micro850 in your Project Organizer, and selecting **Build**.

When the build is complete, you will see a message in the lower left-hand corner that the build has succeeded. If there were errors in your programming, then they would be listed in the **Error List** panel and clicking on the error would direct you to the error in your project.

Now that your build has completed, you can download the project to your controller.
42. Download the project to your Micro850 by clicking the **Download** button on the Micro850 tab.

The **Connection Browser** window will appear. Browse for your controller by expanding **USB** and selecting **16, Micro850, Micro850**, then clicking **OK**.

44. The **Download Confirmation** window will appear. You will be prompted to overwrite the project in the controller. The two choices are **Download** and **Download with Project Values**. When a project is uploaded from the controller, a snapshot of the values of all of the user variables is uploaded and saved. These values can be downloaded back into the controller along with the project by selecting **Download with Project Values**.
45. In this lab, we have a new project and none of the variables have a Project Value assigned to them yet. Therefore, just click **Download**.

46. The download will proceed. When the download is complete, you will be prompted to put the controller back in Remote Run mode. Click **Yes**.

47. Notice that at the conclusion of the download, the software went directly into **Connected** mode, as highlighted below, and is now displaying the real-time status of rung 1. Red contacts indicate that power can flow through them, while blue contacts indicate that the power flow is blocked. Red coils indicate that they are turned on, while blue coils are turned off. Currently the Motor coil is off (blue) because power is blocked by both the blue Start Motor switch contact and the blue Motor ON/OFF contact.
48. Turn and hold the DI0 switch on the demo box. Verify that the _IO_EM_DI_00 Direct Contact has turned red, that the TON timer has timed for one second as indicated by ET=PT=T#1s and that the timer output Q has turned red. Next verify that the _IO_EM_DO_09 Direct Contact and Direct Coil have turned red and remain red, even after releasing the DI0 switch. Finally, verify that the RUN LED on the drive display is on (see picture on next page), although the motor will not be spinning (or it may be barely turning).
49. Turn and release the DI1 switch on the demo box to turn off the output. Notice the drive RUN LED is now off and the corresponding color changes in your Ladder Diagram.

This is a typical motor seal-in circuit (and can also be applied in non-motor circuits as well). The Output Coil is turned on using a Direct Contact and then the active state of the Output Coil seals in the circuit. The circuit is unsealed when a Reverse Contact is turned on.

50. Turn and hold the DI0 switch until the drive RUN LED turns back on, then release DI0.

51. You have completed debugging your motor seal-in circuit – next you will add programming to vary the speed that the motor spins at, while your existing program continues to run.
Exercise #5 - Use Run Mode Change to add a structured text program to control motor speed

In this section, you will learn how to:

- Use Run Mode Change (RMC) to make program changes while the controller continues to run the existing program(s).
- Create a structured text program to control motor speed.
- Archive your project.
You have completed debugging your motor seal-in circuit, but what we really want to accomplish is for the motor to spin when it is turned on and to be able to vary the speed that the motor spins at. The motor currently doesn’t spin when turned on because the Micro850 analog plug-in output point that connects to the drive analog input point needs to be programmed to output something other than 0 volts.

One of the features exclusive to Connected Components Workbench Developer Edition is Run Mode Change (RMC). This feature will allow us to implement program changes without having to stop the process being controlled by the Micro850 controller (i.e., the controller remains in RUN mode).

52. Click on the Run Mode Change icon in the upper right Run Mode Change toolbar. Notice that the rung no longer displays the blue and red status colors and the Toolbox re-appears on the right for programming.

53. Add a Structured Text (ST) program.

Right-click Programs under the Micro850 in your Project Organizer, and select Add ➔ New ST: Structured Text.
54. Notice a new Structured Text program called **Prog1** has been added under Programs.

55. Right-click the **Prog1** program icon and select **Rename**.
56. Rename the program **Speed_Control**.

57. Double-click the **Speed_Control** program icon. A structured text editor will appear in the main project workspace with one blank line. The **Toolbox** contains the structured text command structures.

58. Drag and drop **IF THEN ELSE** from the **Toolbox** into the program space.
Notice that Structured Text is programmed like modern, structured computer programming languages. It is much more efficient than ladder logic for programming complex algorithms and math computations.

The demo kit has a dial labeled **SPEED COMMAND** along the right hand side. This dial is actually a voltage potentiometer and is wired to the first analog input on the Micro850 analog input plug-in module. When the dial is turned clockwise, it increases the voltage at the analog input and when it is turned counter-clockwise, it decreases the voltage at the analog input. The goal of our structured text program is to assign the value received at the analog input to the analog output connected to the drive. That way, turning the potentiometer will vary the drive speed.

59. Therefore, we want the program to assign the value of the analog input #0 to the analog output #1 whenever the motor is running (digital output #9 is on). Replace the (*boolean expression*) in line 1 with `_IO_EM_DO_09`, and replace the (*place you code here*) in line 2 with `_IO_P3_AO_01 := _IO_P2_AI_00;`.

Note that as you type in the variable name, the editor will display the list of variable names that match what you’ve typed in so far. To speed up your programming, you can double click on the variable name in the list instead of typing out the entire name.

60. Delete lines 3-6, since we do not need any alternate (ELSIF or ELSE) code executions. Your 3-line structured text program should appear as shown below. (Make sure a semi-colon is present at the end of lines 2 and 3!)

   ![Structured Text Program](image)

61. Now you are ready to test the changes you made in the running system. Click the ‘Test changes’ icon immediately to the right of **Run Mode Change**.
62. Note that the changes are now built and downloaded to the controller and, once completed, we can monitor how the new program is operating. Click on the Speed_Control-POU tab, then hover over the red underline under _IO_P2_AI_00 to bring up the Click to monitor icon. Click on the icon and adjust the Variable Monitoring window so that you can view the Logical Value of both _IO_P2_AI_00 and IO_P3_AO_01 at the same time. Now when you turn the SPEED COMMAND potentiometer, you will see the value of variable _IO_P2_AI_00 change along with _IO_P3_AO_01, which visibly changes the rotating speed of the motor (with a slight acceleration or deceleration lag). Click Close when you are done.

63. Once the changes have been tested using Run Mode Change, you have two options: Accept changes, which saves the changes that have been made, or Discard unaccepted changes, which deletes the changes that have been made. The icons for ‘Accept changes’ and ‘Discard unaccepted changes’ are immediately to the right of the ‘Test changes’ icon.

64. Now that your changes are verified, click the ‘Accept changes’ icon.
Note again that the motor has continued to spin throughout the Run Mode Change edit session. (When downloading normal offline changes, the controller must be first switched to program mode, which would stop the motor.)

65. Click the **Connected** down arrow and select **Disconnect**.

66. Another feature exclusive to Connected Components Workbench Developer Edition is the ability to archive multiple copies of a project. Especially during development, it can be useful to be able to go back to a previous version of the project if changes that have been made need to be undone. It also provides a chance to document changes from one version of the project to the next. Archive a copy of your current project by selecting **File ➔ Archive Manager**...
67. Enter “Simple motor circuit program and PowerFlex 4M_1 configuration.” for the Description and click Archive.

68. Click Close to close the Archive Manager.
Exercise #6 - Import a project, add a graphic terminal and define tags using direct referencing.

In this section, you will learn how to:

- Import a completed controller project.
- Add a PanelView 800 graphic terminal to the project using ‘Select existing device’.
- Define tags for the PanelView 800 application using direct referencing.
69. Click File ➔ Import Project…

70. Select Motor_Control.ccwarc and click Open.
71. Right click on the Micro850 icon under Project Organizer and click Download.

72. The Connection Browser window will appear. Browse for your controller by expanding USB and selecting 16, Micro850, Micro850, then clicking OK.
73. Click **Download**.

74. The download will proceed. When the download is complete, you will be prompted to put the controller back in Remote Run mode. Click **Yes**.

75. Disconnect from the controller by clicking on the **Connected** down arrow and selecting **Disconnect**.
76. Under **Project Organizer**, click the ‘Add device to the project’ icon.

77. Click **Select existing device**...

78. In the **Connection Browser** window, expand the **AB_ETHIP-1** driver, select the PanelView 800 and click **OK**.
79. Click **Add to Project**.

80. Double-click the graphic terminal icon (labeled as **PV800_App1***) in **Project Organizer** to bring up the PanelView 800 device configuration screen.

**PV800_App1***
81. Double check that the **Landscape** orientation is highlighted then select **OK**.

82. On the **Settings** tab, go to the **Protocol** menu located under **Communication**. Verify that the protocol selected is **Serial | Allen-Bradley CIP** as shown.
83. Leave the communication parameters at their default setting. Notice that the name of the controller we will be using is PLC-1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Controller Type</th>
<th>Address</th>
<th>Timing</th>
<th>Auto-Dem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLC-1</td>
<td>Micro800</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

84. Rename the PV800_App1* as “MotorStarter” by right-clicking on the icon and selecting Rename from the menu.

This application uses read and write tags to define how objects interact with the addresses of the Micro850 controller. You must create these tags before you can assign them to the objects in the application. Only objects that interact with the controller require a tag. Objects such as screen navigation buttons, drawings, and screen text do not require tags.

85. Double-click on the Tags icon to open the Tag Editor.

86. Verify that the External tab is selected.
87. Click the **Add** button to add a tag.

88. Click in the **Address** field and select the (...).

89. Select **Speed_Command** from the list of variables and click **OK**.

90. The **Data Type** is automatically filled in when the tag is selected.

91. Type "**Speed_Command**" in the **Tag Name** field.

**Note:** The Tag Name and Address do not have to match, but it is good programming practice.
92. Click in the **Controller** field and select **PLC-1**. This is the controller name defined on the Communication tab.

93. Repeat the previous six steps to add **Start_Motor** and **Stop_Motor** tags.

94. Add one more tag by selecting the **Add** button.

95. Select the (...) in the **Address** field.

96. Select the **I/O – Micro850** tab.
97. Select the _IO_EM_DO_09 tag and select OK.

![Variable Selector](image)

98. Type “Motor_Ind” in the Tag Name field.

![Motor Ind Tag](image)

99. Click in the Controller field and select PLC-1 and optionally, enter a description.

![Controller Field](image)

100. When done, the Tag Editor should look like this.

![Tag Editor](image)
Exercise #7 - Create a graphic terminal screen, download and test.

In this section, you will learn how to:

- Create a graphic terminal screen with pushbuttons and numeric entry to control the motor.
- Validate, download and test the application on the PanelView 800 terminal.
New applications open with the first screen created. Each screen is created with a default name, Screen_NN, where NN is a number from 1 to 99. The first screen is named Screen_1. You can rename the screen using a maximum of 50 characters.

This is what the Motor Control screen will look like when the following section is complete.

101. Right-click **Screen 1** and select **Rename**.

102. Type "**MOTOR_CONTROL**" into the text field as shown below and press Enter.
103. If the screen tab is not open already, double click on **MOTOR_CONTROL** to open the screen for editing. It should look similar to this:

![Screen Editor Interface](image)

Verify the screen border is highlighted (in red) indicating the screen is selected. You can modify screen properties as long as nothing on the screen is selected.

104. Click on the Properties tab to the right of the Toolbox window to open up the screen Properties window.

![Properties Window](image)

**Note:** The Properties Window can also be activated by right-clicking on the object, in this case the screen, and selecting **Properties**. If you wanted to see the Properties of a push button, right-click on that button and select Properties.
105. Set the Horizontal Grid Spacing and Vertical Grid Spacing to 5, then click on the Properties tab to minimize the window.

![Properties Window]

*Note: From the Properties Window, the Screen Name can also be edited, among other basic settings.*

106. From the Toolbox, locate Momentary Push Button, and drag-and-drop it onto the middle of your screen.

![Toolbox]

*Note: The “Toolbox” is context sensitive. It changes the selection of objects for HMI development versus the controller program development.*
107. Double-click the **button** to open the **States** editor.

Each row is a state with a default value, text, and other format options. Move the scroll bar to see additional options. It should look similar to the one shown below:

![States Editor](image)

108. Edit State 1:

- Click in the **Caption Text** field and type “START MOTOR”

- Click (...) button in the **Background Color** field and select **green** from the **color palatte**.

![States Editor](image)
- Click the (...) button in the **Caption Text Color** field and select **black**.

- Check the **Caption Font Bold** checkbox.

- Change the **Caption Font Size** to “28”.

109. **Edit State 2:**
- Click in the **Text** field and type **“MOTOR STARTING”**
- Select **green** as the **Background Color**
- Select **black** as the **Caption Text Color**.
- Check the **Caption Font Bold** checkbox.
- Change the **Caption Font Size** to “28”.

110. Click **row 1** so that this is the state displayed on the screen in CCW.
111. Verify your **States** window matches the image below and click **OK** to close the window and apply the changes.

![States window](image_url)

112. Make sure the button is still selected and go to the **Properties** window.

113. Configure the following properties as shown below:

- **Height:** 120
- **Left:** 0
- **Top:** 360
- **Width:** 200

![Properties window](image_url)

**Note:** The **Top** and **Left** parameters determine the location of the object. The **Height** and **Width** parameters determine the size of the object.

114. Click the **Write Tag** drop down arrow in the **Connections** section and select **Start_Motor** from the list of tags.

![Write Tag and Connections](image_url)

The **Start Motor** button is complete. Next, create the **Stop Motor** button by using a copy and paste operation.
115. Verify the **Start Motor** push button is selected.

![Start Motor Push Button]

116. Click the **Copy** icon ( 

<table>
<thead>
<tr>
<th>![Copy Icon]</th>
</tr>
</thead>
</table>

) on the toolbar or press the Ctrl+C keys on your keyboard.

117. Click the **Paste** icon ( 

<table>
<thead>
<tr>
<th>![Paste Icon]</th>
</tr>
</thead>
</table>

) on the toolbar or press the Ctrl+V keys.

118. Move the pasted button to the right of the original push button.

![Pasted Buttons]

119. Double click the **new button** to open the **States** editor.

120. **Edit state 1:**
   - Replace the **Caption Text** with “STOP MOTOR”
   - Select red as the **Background Color**
   - Select white as the **Caption Text Color**
   - Double check the **Caption Font Bold** checkbox is checked.
   - Double check the **Caption Font Size** to “28”.

121. **Edit State 2:**
   - Replace the **Caption Text** with “MOTOR STOPPING”
   - Select red as the **Background Color**
   - Select white as the **Caption Text Color**
   - Double check the **Caption Font Bold** checkbox is checked.
   - Double check the **Caption Font Size** to “28”.

122. Click row 1 so that this is the state displayed on the screen in CCW.

123. Click **OK** when done.

![OK Cancel Buttons]

124. Make sure the button is still selected and go to the **Properties** window.
125. Assign tags to the **Stop Motor** push button by clicking the drop down arrow for the **Write Tag**, on the **Properties** window. Assign tag **Stop_Motor** by clicking on it.

<table>
<thead>
<tr>
<th>Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicator Tag</td>
</tr>
<tr>
<td>Visibility Tag</td>
</tr>
<tr>
<td><strong>Write Tag</strong></td>
</tr>
</tbody>
</table>

126. Verify your screen appears as shown below.

The Goto Terminal Config button allows access to the configuration mode screens of the terminal when the button is pressed at runtime. Each application should contain at least one Goto Terminal Config button if the operator needs to access the configuration screens.

127. From the Toolbox, scroll down to the **Advanced** section, and find the **Goto Terminal Config** button. Click and drag it to the center of the screen.

128. Make sure the button is still selected and go to the **Properties** window.

129. We will use the default colors for the object.
130. Configure the following properties as shown below:

- **Font Bold**: True
- **Font Size**: 28
- **Text**: “CONFIG SCREEN”
- **Height**: 120
- **Left**: 400
- **Top**: 360
- **Width**: 200

131. Verify the screen looks as shown below.

The Multistate indicator will show whether the motor is running or stopped. You will be creating the indicator.
132. From the Toolbox, locate the Multistate Indicator object listed in the Display section.

133. Drag the object into the center of the screen.

134. Double-click the object to open the States editor. The indicator is created with five states. This lab only uses two.

135. Delete rows 3 and 4. (Select the row, and click the Delete button). Select OK when prompted.

136. Edit state 1:
   - Accept 0 as the Value.
   - Select red as the Background Color
   - Delete the default Caption Text.
   We will not enter any text since an image will be placed over the multistate-indicator.
137. Edit state 2:
- Accept 1 as the Value.
- Select green as the Background Color
- Delete the default Caption Text.

![States window](image)

138. Click row 1 and then click OK to close the editor.

139. Make sure the button is still selected and go to the Properties window.

140. Configure the following properties as shown below:
- **Height**: 170
- **Left**: 110
- **Top**: 90
- **Width**: 250
- **Read Tag**: Motor_Ind

![Properties window](image)
141. From the Toolbox, locate the **Image** object listed in the Drawing Tools section.

![Drawing Tools](image)

142. Click and drag it onto the screen, directly over the MultiState Indicator.

![Image placement](image)

143. Double-click the “X” to open the image selection window.

144. Click on the **System** tab and locate the **2001R_standard_motor-standard_motor.png**

![Image selection window](image)

145. Click **Select** to apply the picture.
146. Configure the following properties as shown below:

- **Height**: 140
- **Left**: 125
- **Top**: 110
- **Width**: 225

147. Verify the screen looks as shown below.

Note: The image shown in CCW will appear exactly the same on the terminal.
148. The Numeric Entry will allow the user to set the speed of the motor.

149. From the Toolbox, locate the **Numeric Entry** object.

150. Drag the object into the space to the right of the Multistate Indicator.

151. Make sure the button is still selected and go to the **Properties** window.

152. Configure the following properties as shown below:

   - **Font Bold**: True
   - **Font Size**: 28
   - **Height**: 110
   - **Left**: 450
   - **Top**: 120
   - **Width**: 230
   - **Indicator Tag**: Speed_Command
   - **Write Tag**: Speed_Command
   - **Number of Decimal Places**: 1
   - **Numeric Field Width**: 4
153. Verify the screen looks as shown below

Before you run an application on the PanelView 800, it must be validated for errors and warnings. CCW will automatically validate an application once the download is initiated.

154. Right-click on the MotorStarter icon and select Download. Alternately, a Download button is also on the Settings tab.

155. The Validation Results window opens. Any errors listed will need to be corrected before you can download. Close the Validation Results window.

Note: Warnings still may exist, however they can be ignored as they are not functionality errors.
156. Browse for the terminal in the Connection Browser window by expanding AB_ETHIP-1, Ethernet and selecting the 192.168.1.2, 2711R-T7T.

![Connection Browser](image)

157. Then select OK.

158. The Output window will show the progress of the download. If prompted with either “Do you want to stop the loaded application?” or “A file with that name exists. Overwrite?” click Yes. Ensure the application downloads successfully before moving forward.

![Output](image)

159. Turn to your PanelView 800 terminal. It should be on the Main screen.

160. Select the File Manager button.

161. Scroll through the list of applications using the arrow keys to find the MotorStarter application you created.

162. Select Run. If prompted with a warning, select Yes (F1).

163. The application will load. Press the START MOTOR button, and let the motor run. Notice the Multistate Indicator will change color from red to green behind the image of the motor.

164. Select the Numeric Input button you created, enter “10.0” into the keypad, and press Enter. Verify the display on the PowerFlex 4M drive is displaying a number close to 10.0, if not exactly.

165. Press the STOP MOTOR button.

166. Feel free to explore some more with the screen functionality.

167. To exit the application, press the CONFIG SCREEN button. This will bring you back to the File Manager screen.
Exercise #8 - Configure VNC (Virtual Network Computing)

In this section, you will learn how to:

- Use Virtual Network Computing (VNC) to operate graphic terminal screens remotely from your PC.

Virtual Network Computing (VNC) allows you to remotely connect to a PanelView 800 terminal, either over the local network or even over the Internet (with appropriate security in place, of course!). With VNC you can interact with the terminal without having to be physically at the terminal location. It transmits keyboard and mouse actions from your computer to the terminal.
168. Turn to the terminal. Make sure it is on the **Main** configuration screen.

169. Select **Terminal Setting**.

170. Select **Communication**.
171. Select VNC Settings.

![Communication Table]

172. If Status Server shows Disabled, tap Enable/Disable once so that it now reads Enabled.

![VNC Settings]

173. If Status Access shows View-Only, tap View-Only/Control once so that it now reads Control.
174. Set the password for View-Only by selecting **Reset Password** in the **View-Only box**. The default VNC settings (no password) prevents a user from establishing a VNC connection to the terminal.

<table>
<thead>
<tr>
<th>VNC Settings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server:</strong></td>
<td><strong>Enable / Disable</strong></td>
</tr>
<tr>
<td><strong>Status:</strong></td>
<td><strong>Server:</strong> Enabled</td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td><strong>View-Only / Control</strong></td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td><strong>Control</strong></td>
</tr>
<tr>
<td><strong>View-Only:</strong></td>
<td><strong>Reset Password</strong></td>
</tr>
<tr>
<td><strong>Control:</strong></td>
<td><strong>Reset Password</strong></td>
</tr>
</tbody>
</table>

If a user forgets the VNC password there is no way to recover it. The user can perform the restore terminal operation, or return the terminal to the out-of-box configuration. Doing so will clear the password and also remove all applications, logs, recipes, user-installed font files, objects and graphics. Alternatively, you can update the terminal firmware to clear the VNC settings and reset to the default configuration.

175. Enter ‘ra1’ for the **New Password** and **Confirm Password** (and **Old Password**, if shown). Tip - use the tab key to navigate from one password field to the next. Then press the **Enter** key. (The view–only access password can have a maximum of seven alphanumeric characters.) Press **Yes** if confirmation message appears.

176. Set the password for Control by selecting **Reset Password** in the **Control box**.
177. Enter ‘ra’ for the **New Password** and **Confirm Password** (and **Old Password**, if shown). Tip - use the tab key to navigate from one password field to the next. Then press the **Enter** key. (The control access password can have a maximum of seven characters and at least one of them must be one of these special characters: ' _', '$', '-', ' ', or '!'.) Press **Yes** if confirmation message appears.

178. On the computer, launch the **VNC** Viewer by clicking on the VNC icon at the bottom of the screen.

179. Verify the **IP address** of the terminal, 192.168.1.2, and click **Connect**.

180. Enter the password you configured for **View-Only** (ra1), and then click **OK**.

181. The terminal screen is replicated here. Try to navigate on the terminal via the VNC connection. Since you logged in with View-Only rights, you won’t be able to select anything - you should only be able to see what is on the terminal. Now, navigate on the terminal and see the screen update on the computer.

182. Log out of the VNC Viewer by closing the VNC window on your PC.

183. Open another VNC connection by clicking the **VNC** Viewer icon at the bottom of the PC screen.
184. Verify the **IP address** of the terminal, 192.168.1.2, and click **Connect**.

![New TightVNC Connection](image)

185. Enter the password you configured for **Control (ra-)**, and then click **OK**.

![Vnc Authentication](image)

186. Now try to navigate on the terminal via the VNC connection. Since you had the correct credentials, you should be able to control the terminal from your computer. Go back to the **Main** menu and press the **Goto MotorStarter** button to return to your application. Verify that you can start, stop and control the speed of your motor remotely from your PC.

187. When you are done, click on the **CONFIG SCREEN** button to return the terminal back to the **Main** screen. To end the remote connection, simply close the VNC Viewer window on your PC.
Optional Explore PanelView 800 Advanced Features

In this section, you will learn how to:

- Stop a running application on the PanelView 800 terminal.
- Operate an Alarm Summary.
- Change between existing Recipes.
- Login/Logout as different Users.
- View a Trend.
- Uploading existing application from PanelView 800 terminal.

1. On the PanelView 800 terminal, from the Main screen, select File Manager, then select Stop Application.
2. Using the up and down arrows, select **MotorStarter_Advanced** from the **File Manager** screen and select **Run**.

3. From the startup screen, go ahead and start the motor.
4. You should get an alarm banner across the top stating “Motor Started”. (Normally starting a motor would not be considered an alarm condition, but we had to pick something to alarm on!) Select **Close** then **MORE**.

5. Select **ALARMS**.

   ![Motor Started banner](image)

   ![Menu options](image)

   - **Login**
   - **Logout**
   - **Change Password**

   **Current User:**

   ![Menu options](image)
6. Use the up arrow button to highlight the latest alarm, then select the enter button to acknowledge the alarm.

<table>
<thead>
<tr>
<th>Alarm Msg</th>
<th>Occur Time</th>
<th>Occur Date</th>
<th>Ack Time</th>
<th>Ack Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor Started</td>
<td>11:10:10 AM</td>
<td>9/20/2017</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Motor Started</td>
<td>11:04:49 AM</td>
<td>9/20/2017</td>
<td>11:09:31 AM</td>
<td>9/20/2017</td>
</tr>
</tbody>
</table>
7. Select **MOTOR CONTROL** and then stop and start the motor again. This time select **Ack**, then **MORE...**, then **ALARMS** and note that the alarm is acknowledged without having to do it from within the Alarm Summary.
8. Select **MOTOR CONTROL, MORE...**, then **RECIPE**.

9. Use the **Recipe Selector** up, down and enter buttons to select **HIGH_SPEED**, then select **Restore**.
10. If the **Recipe** value is different from the **Current** value, then select **Download** to change the **Current** value to match the **Recipe** value.

11. Select the **NORMAL_SPEED** recipe, **Restore** and **Download** that recipe to the controller.
12. Select **MOTOR CONTROL, MORE..., and Login.** For **User:** type ‘Operator’, press the tab key (highlighted below) to navigate to **Password:**; type ‘123’ and press the enter key.
13. Note that the Current User is now OPERATOR. Select TREND. Note that since OPERATOR does not have the rights to view TREND, you must now login as User: 'Engineer' and Password: ‘pv800’.
14. Turn the **SPEED COMMAND** potentiometer in the demo kit back and forth and see what effect it has on the trend.
15. Select **MOTOR CONTROL, MORE...** and **TREND**, and verify that as long as you are logged in as **ENGINEER**, you can view the trend. (Hint: Select **Logout**, then see what happens when you select **TREND**.)

16. If you would like to view how any of these new screens are configured, you can upload the MotorStarter_Advanced application into a new Connected Components Workbench project by using Discover from the CCW Start Page to browse to the PanelView 800 at Ethernet address 192.168.1.2.

Congratulations – You have completed the lab!