Pico GFX-70 Controllers
Bulletin 1760 Series

Quick Start
Solid state equipment has operational characteristics differing from those of electromechanical equipment. *Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls* (Publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://www.ab.com/manuals/gi) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

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

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

Throughout this manual we use 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.
# Table of Contents

## Chapter 1
Overview of Course 1
- Course 1 Objective ........................................... 1-1
- Task Definition for Course 1 ............................. 1-3

## Chapter 2
Lesson 1
- Creating a Pico Project ................................. 2-1
- Creating a Circuit Diagram ......................... 2-2
- Circuit Diagram Simulation ....................... 2-4
- Establishing a Connection with the Device .... 2-5
- Documenting the Project ............................. 2-7

## Chapter 3
Lesson 2
- Wiring a Pico Controller Timing Relay .......... 3-1
- Timing Relay Simulation .......................... 3-2
- Setting System Properties ....................... 3-3
- Using Notes ............................................. 3-4

## Chapter 4
Lesson 3
- Adding Rungs and Drawing Connections ........ 4-1
- Input Simulation ........................................... 4-3
- Uploading a Program ................................. 4-5
- Using the Cross-Reference List .................. 4-6

## Chapter 5
Overview of Course 2
- Course 2 Objective ........................................... 5-1

## Chapter 6
Lesson 4
- Creating a Pico GFX-70 Project ....................... 6-1
- Program, Circuit Diagram, and Function Block Diagram .. 6-3
- Circuit Diagram and Function Block Diagram Simulation .. 6-7
- Downloading a Program to a Pico GFX-70 Device ........ 6-8
- Printing a Function Block Diagram .......... 6-8

## Chapter 7
Lesson 5
- Pico GFX-70 Pico-Link Project ....................... 7-1
- Programming with Pico-Link Functions ........... 7-6
  - Program for Device 2 (Pico-Link ID 2) .......... 7-7
  - Program for the Master (Device 1, Pico-Link ID 1) .. 7-8
- Pico-Link Configuration and Communication .... 7-11

## Rockwell Automation Support
- Installation Assistance ...................................... back cover
- New Product Satisfaction Return ..................... back cover
Overview of Course 1

Course 1 Objective

Course 1 is designed to help you get to know the PicoSoft Pro user interface. The procedures will acquaint you with the three sections of the user interface and the four different views that you will use. Take advantage of this course to see how simple it can be to work with PicoSoft Pro.

Course 1 consists of:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Lesson 1</td>
<td>Creating a Pico Project</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Creating a Circuit Diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Circuit Diagram Simulation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Establishing a Connection with the Device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Documenting the Project</td>
</tr>
<tr>
<td>3</td>
<td>Lesson 2</td>
<td>Wiring a Pico Controller Timing Relay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Timing Relay Simulation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Setting System Properties</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using Notes</td>
</tr>
<tr>
<td>4</td>
<td>Lesson 3</td>
<td>Adding Rungs and Drawing Connections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Simulation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uploading a Program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using the Cross-Reference List</td>
</tr>
</tbody>
</table>

Course 2 provides more advanced information on using the features found in PicoSoft Pro.
After completing Course 1, you should be able to:

• Create a project,
• Wire up a circuit diagram,
• Test a circuit diagram,
• Transfer it and
• Print it out.

To make optimum use of PicoSoft Pro, you must ensure that your screen is using a resolution of 1024 x 768 pixels and small fonts (96 dpi). The necessary settings can be made via Start, Settings, Control Panel, Display, Settings.

When switching on a Pico controller for the first time, you must first select the language before working with the device or connecting PicoSoft Pro.

The language selection menu is shown in the Pico display. Select your language using the arrow buttons on the keypad. Confirm your selection with OK or exit the menu via ESC.
Task Definition for Course 1

The example procedures will refer to the following system.

A conveyor belt system that is driven by a three-phase motor is to start up with a delay of 3 seconds after it has been switched on. The conveyor belt is used for transporting packages. Once a certain number of packages have been transported, the system should switch off after a specified rundown time.

In our example, the number of packages to be transported is 5 and a rundown time of 4 seconds is set. The system is switched on via S1. S2 switches off the system immediately and light barrier, S3, is used to detect the number of packages that have already been transported.

**Inputs**

- S1: Switch on system (make contact)
- S2: Switch off system (break contact)
- S3: Light barrier (make contact)

**Outputs**

- K1: Contactor for switching the three-phase motor

**Parameters**

- T1: Startup delay (3s)
- T2: Switch-off delay (4s)
- C1: Number of packages (5 pieces)
The connection diagram is as follows:

Proceed to Lesson 1.
Lesson 1

Creating a Pico Project

To create a circuit diagram in PicoSoft Pro for a Pico controller, you first have to create a project. The project consists of the device (e.g. 1760-IB12XOB8) and the associated circuit diagram. You can start a new project by left-clicking File then New.

The PicoSoft Pro user interface is divided into three sections. The following figure shows you the user interface in the Project View. This View consists of a Toolbox [1], a Properties field [2] and the Workbench [3].

Before we can start wiring up the circuit diagram, we have to select a device from the Toolbox [1] and add it to the Workbench using drag and drop. To create a solution for our task definition we need a 1760-L12AWA controller.

We can select this device from the Toolbox in the following way:

1. Clicking the “+” to the left of the heading “1760-L12 Series” will open the tree structure to show a list of devices in this series.
2. Click the device called “1760-L12AWA”, hold down the left mouse button, and drag the mouse (now showing a device symbol) to the right onto the Workbench.

The Properties field [2] now shows an overview of the device properties. Right-click the device and choose Device Information in the context menu. The properties of the device include, for example, the number of inputs and outputs, the number of markers, timing relays, and counters. Use this field to verify that you have chosen the right device before creating the circuit diagram for your application.

If necessary, delete an unsuitable device using Delete Device in the context menu.

Creating a Circuit Diagram

Once we have selected a device in the Project View, we can now switch to Circuit Diagram View. To do this we double-click the device.

Like the Project View, the Circuit Diagram View is also divided into three sections:

This view also consists of the Toolbox [1], the Properties field [2] and the Workbench [3], which is also called the Circuit Diagram window in this case.

To wire up the circuit diagram, we need operands (e.g. I inputs, Q outputs, M markers, etc.). The operands available are displayed in the
Toolbox [1] and can be logically linked together in the Circuit Diagram window [3]. You can specify the properties of the operands in the Properties field [2].

To develop a solution for our task, we first have to wire up the inputs I1 and I2 in our circuit diagram so that they act on marker M1, which in turn is responsible for the on/off switching of the motor. To use input I1 in our circuit diagram, proceed as follows:

1. Left-click in the Toolbox [1] the operand called “I Input Basic Unit” and hold down the mouse.

2. Drag the mouse, which will then change to the operand symbol for the I input, to the right onto rectangle 1A in the circuit diagram [3] and release the mouse button.

3. Select an operand number (in this case 1) from the I list box (on the right of the “I”) in the Properties field [2]. As it is a local input of this device, the 0 shown in the Station list box is unchanged.

You can also assign a comment to the operand (e.g. Motor on), if required. Your entry is accepted immediately.

You can then add all the other operands to the circuit diagram in the same way so that it will resemble the following figure:

To ensure a fail-safe circuit diagram, I2 has been wired as a break contact.

**TIP**

You can place operands in the circuit diagram wherever the mouse pointer takes on the appearance of the operand symbol. When it looks like the 'Prohibited/No Entry' sign, this indicates that the area is reserved for a connection line.
Circuit Diagram Simulation

You will need to select Simulation View to test the wired circuit diagram.

Activate the Simulation View via View, Simulation or by clicking the Simulation button at the bottom of the Toolbox. The following figure shows the Simulation View in PicoSoft Pro.

The Simulation View is also divided into the three areas of the PicoSoft Pro user interface. The Toolbox [1] contains the simulators for the inputs and different dialogs for setting simulation parameters. The Properties field [2] takes on the role of a display panel on which you can watch the states of inputs, outputs and markers. The circuit diagram area [3] is used to display the power flow of the different rungs of your circuit diagram.

To test our circuit diagram, proceed as follows:

1. Left-click the Play button \( \rightarrow \) in the toolbar to start the simulation.

2. Click the I Inputs button in the Toolbox. This will open a dialog that displays the physical inputs of the device.

3. Click the switches of inputs 1 and 2 in the Toolbox [1] and watch the behavior of the power flow in the circuit diagram [3]. Energized rungs are highlighted in red.
4. For example, to view the status of a marker in the Properties field window, simply double-click the marker you wish to monitor.

5. Click the Stop button to stop the simulation.

Establishing a Connection with the Device

Now that we have tested the circuit diagram, let's download it to the device. To do this, connect your device to the PC using the 1760-CBL-PM02 cable available as an accessory.

Choose Communication, Interface, select the interface required (COM 1 ... 4), and then click the Communication tab to move to the Communication View.

**TIP** Ensure that no other application is currently running on the interface you have selected.

When you move to Communication View, PicoSoft Pro will immediately try to establish a connection to the device. The Communication View will appear without any error messages if this is successful.

If, however, the connection fails, a corresponding error message will appear, although the Communication View will still open. You must then check the connection between the device and the PC once more and, if necessary, select another COM interface. Choose Communication, Online to start another attempt to establish a connection to the device.

The Communication View is also divided into three sections. It looks very much like the Simulation View and contains the Toolbox [1], a Properties field [2] and a Circuit Diagram [3] like the last two views described. The “Communication View” with its three divisions is shown below.
Don’t worry if the circuit diagram suddenly appears empty. This area is used to display the current state of the circuit diagram when it is running in the device. More about this later.

First, we need to transfer the circuit diagram to the device, and then start the device. The Toolbox [1] will help us to do this. Proceed as follows:

1. Click the Program button in the Toolbox. This will open the dialog for downloading and uploading the circuit diagram.

2. Click the Download button. The circuit diagram will be transferred if the device is in Stop status. If this is not the case, a dialog will appear via which you can stop the device and proceed with the download process. The Properties field window [2] shows the progress bar via which you can see the current status of the data transfer.

Once the download has finished, we can then display the current status of the circuit diagram, the so-called “Power flow”, in the device:

1. Start the device by clicking the RUN button in the Program dialog of the Toolbox [1]. Alternatively, you can choose Communication, Device RUN.

2. Click the Play button. The power flow will start and the current states in the device will be displayed.
3. Click the Stop button to stop the Power flow display.

You have now been shown the four views of PicoSoft Pro and are now able to:

- Select a device for the project (in Project View)
- Create the circuit diagram (in Circuit Diagram View)
- Test the circuit diagram (in Simulation View)
- Transfer the circuit diagram to the device (in Communication View)

We will now learn how to document the project.

**Documenting the Project**

To create documentation for the entire project (devices and circuit diagram) we must move to Project or Circuit Diagram View.

In our example let us select the Project View via the context menu, by clicking the Project button in the Toolbox, or alternatively, via Project, Project View.

1. Click File, Form Setup... to open an editor dialog that will help you design the look of the footer text.

   This is where you can assign entries for the document number, date etc. to the form layout. If required, you can also add a company logo or picture to the footer text in the form of a *.bmp graphic file.

   The *.bmp graphic should be 220 x 70 pixels in size for optimum display.

2. Click OK to confirm your entries and exit the dialog. Click File, Print... to set the print range, and then click the OK button to start the printout of the documentation.

   The format of the printout depends on the display format set in Circuit Diagram View (Device, DIN IEC or ANSI/CSA).

Proceed to Lesson 2.
Lesson 2

Wiring a Pico Controller
Timing Relay

Back to our task. In order to extend the circuit diagram for our conveyor belt system we have to switch to Circuit Diagram View.

As we know from Lesson 1, there are several ways to change views. We can either use the menu (Circuit Diagram, Circuit Diagram View) or the tab in the Toolbox. However, there is a more convenient and faster way of reaching this view. Double-clicking the device is enough to activate the Circuit Diagram View.

Since the conveyor belt is required to start up with a delay after being switched on, we shall use the T1 timing relay to create the on-delayed function. The task definition requires a time delay of 3s.

T1 can be added to the circuit diagram as follows:

<table>
<thead>
<tr>
<th>Circuit Diagram Element</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>T:</td>
<td>Comment:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Contact
  - Make Contact
  - Break Contact

- Coil Function
  - Trigger
  - Reset
The T1 contact activates coil Q1 once the delay time has elapsed. The parameters for T1 are assigned when the function relay is entered:

### Timing Relay Simulation

PicoSoft Pro does not have an “Autosave function”, so it is advisable to save the circuit diagram you are creating at regular intervals (File, Save).

**Timing Relay Simulation**

To check the extended circuit diagram, choose View, Simulation to move to the Simulation user interface.

In this lesson we will observe the switching behavior of timing relay T1. To do this we will display the timing relay in the Properties field window by:
1. Clicking the **Display** button in the Toolbox,

2. Double-clicking the text **T Timing Relays**,

3. And then clicking **T 1-2** to switch to the Properties field.

Now, start the simulation by clicking the **Start Simulation** button.

You can now actuate switches I1 and I2 in the Simulator for the I inputs so that the timing relay T1 is started. The output Q1 will then be switched on after the on-delay of 3 seconds has elapsed.

Close the simulation via **Simulation, Stop**.

---

### Setting System Properties

In this lesson, we are going to modify the system settings of the device “online”.

1. In our example, let us select the Communication View via the context menu, by clicking the **Communication** button in the Toolbox, or alternatively, via **View, Communication View**.

   Once Communication View has been opened successfully, we can immediately start to modify the system settings of the device.

2. Click the **System Settings** button in the Toolbox.

   PicoSoft Pro will display the system settings that you can change. These include the enabling/disabling of P buttons and the menu language selection for the device.

   Experiment with the check boxes and watch the effects in the status display of the device. If the check boxes are grayed out so that you cannot select them, this means that there is no communication established between PicoSoft Pro and the device.

3. Check your connection to the device. Is the device connected to the PC via the **1760-CBL-PM02** programming cable? Is the COM interface number correct?

4. Re-establish an online connection. To do this, choose **Communication, Online**, or alternatively, via the **Connection** dialog and the **Online** button in the Toolbox.
PicoSoft Pro enables you to attach notes to individual rungs to add comments to your circuit diagram. To use these freely definable comments, switch to Circuit Diagram View (for example, via View, Circuit Diagram).

We are now going to attach a note to our circuit diagram. To do this, proceed as follows:

1. Left-click the Note button in the toolbar. The mouse will change to the shape of a Note.

2. Position your mouse between two rungs (e.g. 1 and 2) and place the note by left-clicking the mouse. The entire writable range is highlighted in color.

3. The Properties field will then show a text entry area in which you can enter the comment (e.g. Test). Click Apply to transfer the entered text to the note.

Working with notes:

- If the notes get in the way when you are wiring the circuit diagram, you can show or hide them as required by choosing View, Note.
- You can click and drag a note to a different rung.

TIP: The connection to the device will be cleared as soon as the Communication View is closed. PicoSoft Pro will therefore deactivate the menu items in the Communication menu. The only exception to this is the COM interface selection.

TIP: Notes must be visible (factory setting) otherwise use the button to make notes visible.
• To delete the note, position the cursor on the note field, right-click the mouse, and choose **Delete Note**.

• If the notes are visible, they will be printed out with the circuit diagram (**File, Print...**).

The toolbar features the Note button and four other tools for editing your circuit diagram:

• **Cross-reference list**, for displaying this list in the **Properties field** window.

• **Select Area** for selecting areas of the circuit diagram for deleting or copying.

• **Pen**, for drawing connection lines between contacts, coils, function relays and function blocks.

• **Eraser**, for deleting connection lines, contacts, coils, function relays and function blocks.

After becoming familiar with the toolbar, proceed to Lesson 3.
Lesson 3

Adding Rungs and Drawing Connections

Now let us focus entirely on our solution. Light barrier S3 supplies count pulses via I3 for the packages that are transported on the conveyor belt. These are connected to the C input of Counter 1 for as long as the motor is running. The Reset condition for Counter 1 is present when the motor stops (Q1 = 0). The extended circuit diagram looks as follows:

![Circuit Diagram](image)

Remember that the conveyor belt should stop after the maximum number of transported packages has been reached. As soon as the setpoint value of 5 is reached, the output of Counter 1 is set and thus supplies the required signal. The counter is assigned parameters in a similar way to timing relay T1. The following figure shows the Properties field for the counter relay parameters.
Up to now, only I2 was used to stop the motor manually. Both manual and automatic stop conditions can be implemented by wiring the counter output as a make contact parallel to I2.

To do this we add an empty rung between rung 2 and 3. This is done by positioning the cursor on rung 3 and adding a new rung via Edit, Insert Rung. The counter contact 1 is then linked as follows:

![Circuit Diagram](image)

To draw a vertical connection line between rung 2 and rung 3, you need the Pen from the Toolbox. You can activate the Pen in the same way as the Note function (see Lesson 2, Documentation). You can also position the mouse on the circuit diagram and right-click to open the context menu. You can then choose the Pen function and disable it after completing the drawing.
A rundown time of 4s after the switch off time is required to allow all packages to completely come off the conveyor belt. This delay is provided by T2 which activates the switch off as an on-delayed timing relay. For this we shall modify the circuit diagram as follows:

The output of C1 triggers T2 which then controls M1 in rung 3 instead of C1.

**Input Simulation**

You can test the circuit diagram easily by adapting the function of the input switches to the requirements of the task definition. I1 and I3 should be actuated by momentary make contacts, I2 by a momentary break contact.

The function of the individual input simulators can be set in the Simulation View. To do this switch to Simulation View and click the I/R Function button.

This will open the I/R Function dialog in which you can carry out the necessary operations for our exercise.
Once the settings described above have been made, you can then start the simulation by clicking the Play button in the Toolbar. Watch the effects of the modified functions of the I input simulator.

You can then end the simulation by clicking the Stop button.

Changes in the I/R Function dialog change the switching characteristics of the I or R input simulators.

1: “Momentary make contact” function

2: “Momentary break contact” function
Uploading a Program

We will now “upload” a circuit diagram from the device. PicoSoft should be able to create the project automatically for us.

1. Save your project via **File, Save As...**. This will open a standard Windows file save dialog. Give the project a suitable name (e.g. “conveyorbelt”) and save it via the **Save** button. The project will be saved with the file suffix **e40**.

2. Now close the project via **File, Close** (not **Exit**).

3. This will cause PicoSoft to close the project you have just worked on and immediately open an empty project called untitled<Number>.e40.

   Alternatively, you can open an empty project by clicking **File, New** or click the New button directly on the toolbar. This will activate a second instance of PicoSoft and minimize the last edited project so that it is shown in the taskbar.

4. We will switch to the Communication View directly without transferring a device to the Workbench by choosing **View, Communication**.

The following dialog will appear:

```
Device Series

You must specify the device series in order to set up communication with the connected device.

Type of Connected Device:

- Device L1x
- Device DUx

OK  Cancel
```

5. Now let PicoSoft know which device you have connected to the PC. As we have been using the 1760-L12AWA device from the 1760-L12 series, select **Pico Device** and confirm your selection with **OK**. You will automatically enter the Communication View.

Don’t worry if you have set the wrong device series by mistake. After acknowledging the resulting error message, you can still enter the Communication View. You can now redefine the connected device via **Communication, Interface, Pico Device**.
6. Re-establish an online connection to the device (menu: Communication, Online).

7. Once the connection has been established successfully, you can then transfer the circuit diagram into PicoSoft via Communication, Program, Upload.

8. If the circuit diagram upload was successful, you can view it in the Circuit Diagram View (View, Circuit Diagram) and see the associated device in the Project View (View, Project).

Using the Cross-Reference List

The cross-reference list allows you to obtain a fast overview of the operands used in the circuit diagram. If you have not opened your project, open the one you have previously saved by choosing File, Open so that we can view a cross-reference list.

1. Select the appropriate PicoSoft project file, which has the suffix e40. In our example, the file is called “conveyorbelt.e40”. Click the Open button.

   PicoSoft will load the conveyor belt project.

2. Switch from Project View to Circuit Diagram View via Circuit Diagram, Circuit Diagram View.

3. Once you are in Circuit Diagram View you can now display the cross-reference list. To do this, choose Circuit Diagram, Cross-Reference List or alternatively, the cross-reference list button in the toolbar.

| Cross-Reference List |

| Cross-Reference List [1]: Press Enter to change one of the comments. |

<table>
<thead>
<tr>
<th>Operand</th>
<th>Circuit Connection</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1 1A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2 2A, 5A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q1 4G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M1 1G, 2G, 3A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1 3G, 4A</td>
<td></td>
<td>Emergency-stop</td>
</tr>
<tr>
<td>C1 5G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can also use the cross-reference list to assign comments.
4. Left-click a comment field (in our example the comment “Emergency-stop” was assigned to I2).

5. Type your comment and press Enter. This will assign the comment to the operand directly in the circuit diagram.

6. The cross-reference list can be printed out via File, Print... and thus added to the project documentation.

When you are comfortable with the information presented in Course 1, proceed to Course 2.
Overview of Course 2

Course 2 Objective

Course 2 assumes the learner already has a general knowledge of using PicoSoft Pro. If this is not the case, please go through Course 1 first.

The exercises in Course 2 primarily describe how to create a project for Pico GFX-70 devices. However, users of other Pico controllers can also take part in this course to get acquainted with the wide range of functions available with the Pico GFX-70 devices.

The course consists of two additional lessons. **Lesson 4** covers a project using a Pico GFX-70 device as a stand-alone device. It will show you how to wire up function blocks, explain the function block diagram and circuit diagram, as well as showing how to simulate function blocks.

**Lesson 5** primarily deals with creating projects containing several Pico GFX-70 devices. These devices can be connected up via the “Pico-Link” intraconnect, enabling them to communicate with each other. The lesson shows you how to create this kind of master-slave project, how to configure it and how to download several programs to the individual devices, and also describes program uploads and the status display in PicoSoft Pro.

Take some time to go through this course as well. It will explain simply and clearly how to create Pico GFX-70 projects and will provide the answers to a large number of questions from the start.
Lesson 4

Creating a Pico GFX-70 Project

We learned how to create a project in Lesson 1 (reference: Creating a Pico Project on page 2-1).

We are now going to create our first project which has remote expansion I/O as well as the Pico GFX-70 basic unit.

To create a project:

1. We shall start a new project via File, New.

2. We then open the device tree for the 1760-DU series in the Toolbox by double-clicking the Device icon, or clicking the “+” sign.

3. We then add a 1760-LDF CPU module, a 1760-DUB display module and a 1760-IB12XOB4IOF I/O module to the Workbench using drag and drop.

We need an expansion module to locally expand the basic unit. This module can be placed in the Workbench in the same way as the basic unit.

1. Scroll down the Toolbox until you see the expansion modules.

2. Select the 1760-IB12XO8 expansion module in the Toolbox, hold down the left mouse button and drag the module onto the Workbench. Your project should now resemble the following figure.

3. To complete the project configuration, the system properties of the device must also be specified. Click the basic unit to show the device properties in the Properties field window.
You can then make all the necessary settings in the appropriate tabs. The settings will be downloaded to the device together with the circuit diagram.

4. The **Mode** tab should be the first tab to be activated. The **Mode** area indicates at **Display** that this first basic unit is initially configured for stand-alone operation.

5. Select the **System** tab. In our example we want to set the following device properties for the Pico GFX-70 device in the **System** and **Daylight Saving Time** areas:
   - Debounce active
   - P Buttons active
   - Retention On for: MB 1 (Marker byte 1 -> Marker bits 1-8) and C 1 (counter function block)
   - Circuit Diagram Name: Device_1
   - Daylight Saving Time: EU (the automatic time change is based on the DST for Central Europe).

   **TIP**  
   **Note on retention setting:** The setting 1 - 0 has the same effect as 1 - 1, and therefore only declares retentive the first operand stated.

When the **System** tab is activated, the **Properties field** then looks like this:

<table>
<thead>
<tr>
<th>Mode</th>
<th>System</th>
<th>Security</th>
<th>Communication Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Image of the Properties field" /></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Let us delete the expansion unit as we won't need it in the next exercises. To do this position and right-click the mouse on the expansion device and choose **Delete Device**.

**TIP**  
Deleting the basic unit will automatically delete the circuit diagram that belongs to it.
Let us switch to Circuit Diagram View and look at the operands provided in the Toolbox. As you can see, there are a number of new operands available for programming Pico GFX-70 devices.

As with Pico projects, the circuit diagram contains contacts and coils that are linked together via connection lines. These contacts and coils can, for example, be the inputs and outputs of the controller. However, function blocks are also provided with contacts and coils that you can use in the circuit diagram to evaluate the function block concerned, or control it in some way.

To illustrate the interaction between the function blocks and the circuit diagram, let us write a small and simple practice program.

The aim of the program is to count the number of actuations on the P1 button of the device using the counter function block (C..), and to display the counter status in binary format on the outputs Q1-Q4.

We will use the values in the following table:

<table>
<thead>
<tr>
<th>Counter status</th>
<th>Q4</th>
<th>Q3</th>
<th>Q2</th>
<th>Q1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

If the P button is pressed for the 15th time, the counter (C..) and the outputs should be reset to zero the next time that P1 is pressed.

To implement this task we first need the input signal of the P1 button. This is passed to the counter C1 which saves each actuation signal from P1.
1. Drag the P Button operands and counter (C..) in succession from the Toolbox, drop them onto the circuit diagram and arrange them as shown in the following figure.

2. The Properties field window in the Mode tab will propose the operand number 1.

   We will accept the proposed number 1.

   The first line of our program will look as follows:

   ![Circuit Diagram](image)

3. As the counter function block is to count the rising edges only, we must select the option C_ counter coil and Rising edge in the list box.

4. Select in the Properties field window the Parameters tab.

5. We assign the function block with setpoints in the Function Block Inputs field. In the SH area for defining the upper setpoint we select NU (Number = constant value) in the left box and then enter 16 in the entry box on the right.

6. We want to save the counter result in marker byte 1 (contains the marker bits 1-8). In the Function Block Output field we thus have to assign the word output QV to MB 1.

   ![Parameters Table](image)

7. In the Parameter Set list box, select the setting Call Enabled.
This enables the parameter set of the function block to be called and modified on the device during operation. For this, the program must have been transferred to the device beforehand.

We have now wired the most important function block in our mini project and defined its parameters. However, the counter function block is then not only displayed in the circuit diagram. Unlike Pico projects, Pico GFX-70 projects also have a function block diagram as well as the circuit diagram. We can see this via View, Function Blocks or directly via the button.

The function block diagram provides the user with an overview of all the function blocks used in the program as well showing which of the inputs and outputs of the function block are used where.

We have used counter C01 in our program, set its upper setpoint (SH) to 16 and buffered the actual value in marker byte 1. We have also assigned the P1 button to its Count coil so that the function block knows when to increment its actual value by 1. The counter is shown in the function block diagram in the following way:

A function block can contain inputs and outputs as well as contacts and coils. The inputs and outputs of the function block are always shown in bold type (in this case SH, SL, SV, and QV). The inputs are located on the left and the output on the right. The function block inputs read 32-bit values, and the QV output also supplies the actual value of the function block as a 32-bit value. QV can be saved in a marker byte (MB), marker word (MW), marker double word (MD) or passed on directly to the output of the device.

The function block contacts are shown in normal type on the right of the function block. They can be integrated in the circuit diagram as contacts. These contacts can also be used to read the states of the function block and logically link them in the circuit diagram.

Example: In our project, when the counter has reached 16, this sets function block contact OF (actual value greater than or equal to upper
setpoint SH). We can use this signal in the circuit diagram to reset marker byte 1 to zero.

The function block coils are located on the left of the function block and are also shown in normal type. These coils are needed for activating a function block or supplying it with count pulses. We have already implemented the supply of the count pulses via C_.

If you have used any contacts or coils in the circuit diagram, this is highlighted in the function block with green circles 🔄.

To complete the wiring of our circuit diagram let's move back to it (menu: View, Function Blocks or via the 🔄 button).

The aim of our task is to visually display the actual value of the counter using outputs Q1-Q4. As the actual value is stored in marker byte 1, we can use marker bits 1-4 (M1 - M4) for controlling the outputs (see the following table).

<table>
<thead>
<tr>
<th>Applicable to MD, MW, MB, M</th>
<th>Left = Most significant bit, byte, word</th>
<th>Right = Least significant bit, byte, word</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-bit</td>
<td>MD1</td>
<td></td>
</tr>
<tr>
<td>16-bit</td>
<td>MW2</td>
<td>MW1</td>
</tr>
<tr>
<td>8-bit</td>
<td>MW4</td>
<td>MB3 MB2</td>
</tr>
<tr>
<td>1-bit</td>
<td>MB4</td>
<td>MB3 MB2 MB1</td>
</tr>
<tr>
<td></td>
<td>M32...M25</td>
<td>M24...M17 M16...M9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>M8...M1</td>
</tr>
</tbody>
</table>

The control of Q1-Q4 is implemented as follows:

![Control diagram](image-url)
What is now still missing is the resetting of the counter when the actual value has reached 16. We also want to zero the wired outputs. To do this, we use the function block contact OF of the counter.

As previously described, OF is set to “1” if the counter actual value is greater than or equal to the upper setpoint. OF then switches the reset input of the counter as well as activating the MR - Master Reset function block, which, in turn, clears marker byte 1.

The last two lines of the circuit diagram are shown in the following figure.

The Master Reset MR1 function block is set so that it clears the marker double words 1-48. Our marker byte 1 is thus also reset and outputs Q1-Q4 are set to “0”.

Before downloading our program to the connected device, let us look at it and check it in the Simulator.

1. To do this select Simulation View, for example by choosing Simulation in the context menu.
2. To view the status of the outputs, click the Display button in the Toolbox and click the Q Outputs tree structure. The Properties field window shows the outputs as round grey symbols.

3. To view the states of the function blocks as well as the outputs, we have to switch to function block diagram via menu: View, Function Blocks or directly via the button.

The simulation can also be started by clicking the button in the toolbar.

4. Then open the P Buttons dialog and click on the button for P1. Observe the response of the outputs Q1-Q4 as well as the status changes in the function blocks.

**Downloading a Program to a Pico GFX-70 Device**

The program download to a Pico GFX-70 device is similar to the download to a Pico controller:

1. Ensure that your PC is connected to the device and switch to Communication View.

2. Open the Program dialog in the Toolbox and click the Download button.

The program is transferred to the device.

3. Once the transfer has been completed, start the device via the Program dialog and Run.

4. You can watch the online power flow display by clicking the Start button in the toolbar.

5. Press the P1 button on the device and check whether outputs Q1-Q4 are switching correctly.

**Printing a Function Block Diagram**

A Pico GFX-70 project has a function block diagram and a circuit diagram that can be printed out for the documentation.

Select Circuit Diagram View to print the function block diagram.

1. Call the Print and Page Preview Contents dialog via File, Print....
2. Select Program 1 and Function Blocks 1.

3. Click the OK button to send the print job to the printer.

Proceed to Lesson 5.
Lesson 5

Pico GFX-70 Pico-Link Project

When using Pico GFX-70 devices, you can connect several basic units. This connection is implemented using Pico's **Pico-Link**. This CAN-based intraconnect allows up to 8 devices to communicate between each other. The entire Pico-Link intraconnect can be up to 1000 meters in length.

A Pico-Link intraconnect consists of at least two basic units, with the first one being the master and the second one being the slave. The maximum configuration possible consists of one master and seven slaves.

In this lesson we are going to create a Pico GFX-70 project consisting of one master and two slaves. For this, you must be in Project View. The first slave (intelligent slave) is to process a circuit diagram. The second slave is only meant to operate as a remote I/O module, in other words, without its own circuit diagram.

The following figure shows the required project configuration.

To create the project shown above, proceed as follows:

1. Open a new project via **File, New**.

2. Double-click **1760-DU series** on the left in the Toolbox.

3. Then add a **1760-LDFC** CPU module, a **1760-DUB** display module and a **1760-IB12XOB4I0F** I/O module to the Workbench using drag and drop. Confirm any prompts that appear by clicking **OK**.

4. Repeat this configuration twice more for the other two devices.
Since PicoSoft Pro sets everything automatically for you, that's all you have to do.

But what did PicoSoft Pro do for us? Let's start once more from the beginning to examine the steps in more detail.

1. Close the current project via **File, Close** without saving it.

2. Open a new project via **File, New**.

3. As described above, we then add a **1760-LDFC CPU module**, a **1760-DUB display module** and a **1760-IB12XOB4IOF I/O module** to the Workbench using drag and drop.

   After it has been placed on the Workbench, this first device will be shown without any Pico-Link ID.

4. Repeat the above configuration procedure for the next device.

   The following Pico-Link ID dialog will appear after you have dragged the processor onto the Workbench and released the mouse button.
5. You will be asked to enter a Pico-Link ID (station number). PicoSoft Pro will propose the Pico-Link ID = 2.

After entry, you will be asked to enter the device version number and confirm the Pico-Link ID = 2 via the OK button.

PicoSoft Pro will now create a Pico-Link intraconnect. The first device is automatically assigned Pico-Link ID 1 and is “promoted” to master. The second device is assigned Pico-Link ID 2 and thus automatically becomes a slave.

In a project with more than one basic unit, the first device is always the master (top left) and is thus assigned Pico-Link ID 1. Every additional device is assigned a Pico-Link ID greater than 1 and can never be the master.

The Pico-Link Configurator area now looks as follows:

The MASTER is set to Pico-Link ID 1.

The Pico-Link Configurator area shows that the Pico-Link ID has been set to 1. This setting cannot be changed since there must always be one master.

You can also use Send IO to specify whether each status change of the inputs and outputs should be sent to the other Pico-Link stations.
PicoSoft Pro primarily uses the term Pico-Link ID for station number. In other words: Pico-Link ID = Station number.

2) With the SLAVE (Pico-Link ID 2)

The Pico-Link Configurator area for the slave looks quite different. Here you can select or deselect both Send IO and Remote RUN. The selection for Remote IO always remains inactive and is automatically determined by PicoSoft Pro. The following table shows an overview of the configuration settings and their meanings.

<table>
<thead>
<tr>
<th>Description</th>
<th>Master (Pico-Link ID=1)</th>
<th>Slave (Pico-Link ID=2...8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote IO</td>
<td>Cannot be set, is always disabled</td>
<td>Is automatically determined by PicoSoft Pro, depending on whether a program was created for the Pico-Link station.</td>
</tr>
<tr>
<td>Remote IO basically means here decentralized, remotely located inputs and outputs (I/O device).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If Remote IO is set, this is a slave that does not contain a circuit diagram. The device is used as a remote input/output expansion module (I/O device). This is therefore a “dumb” slave.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If a program is created in Circuit Diagram View for this slave, Remote IO will be immediately disabled. The “dumb” slave becomes an “intelligent” slave.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If a slave is to be used only as a simple input/output module, it must be ensured that it does not contain a circuit diagram. In this case, the outputs of the remote IO slave can only be set by the master. The inputs/outputs of a remote expansion module can, however, be read by any “intelligent” device in the intraconnect.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
After dragging the second device on the Workbench, you will have noticed that the device has two additional setting modes displayed above it for setting the Pico-Link. This is the transfer speed (Baud Rate) and the bus delay time of the Pico-Link (Bus Delay). The following figure shows the control elements via which you can determine the speed of the Pico-Link.

<table>
<thead>
<tr>
<th>Description</th>
<th>Master (Pico-Link ID=1)</th>
<th>Slave (Pico-Link ID=2...8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send IO</td>
<td>Can be set</td>
<td>Can be set</td>
</tr>
<tr>
<td></td>
<td>If the inputs/outputs of another device are to be made accessible to the other Pico-Link stations, <strong>Send IO</strong> must be activated. In this case, the states of the inputs and outputs are made available cyclically to the Pico-Link intraconnect and thus to the other Pico-Link stations. Input/output devices must always have <strong>Send IO</strong> activated so that station 1 is always provided with the latest input and output data. If Send IO is activated, the volume on the intraconnect can be considerably increased. This will lengthen the reaction times.</td>
<td></td>
</tr>
<tr>
<td>Remote RUN</td>
<td>Cannot be set, is always disabled.</td>
<td>Can be set</td>
</tr>
<tr>
<td></td>
<td>Activate <strong>Remote RUN</strong> if the possible stations 2 to 8 are to automatically follow the RUN/STOP operating mode changes of the master. <strong>WARNING</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If several engineers are commissioning a machine or system involving several spatially separated elements via the Pico-Link, it must be ensured that <strong>Remote RUN</strong> is not activated. This may otherwise cause uncontrolled switching states during commissioning in the machine or system concerned. <strong>Serious injury or damage</strong> may occur depending on the installation concerned.</td>
<td></td>
</tr>
<tr>
<td>Pico-Link ID</td>
<td>Cannot be set, always = 1</td>
<td>Can be set between 2...8</td>
</tr>
<tr>
<td>1 = Master</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2...8 = Slave</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Pico-Link ID is also called the station number and can be assigned irrespective of the physical location of the device in the intraconnect.</td>
<td></td>
</tr>
</tbody>
</table>

Baudrate: [125 KBaud]  Bus-Delay: [0]
We shall not, however, discuss the possible settings of Baud Rate and Bus Delay any further at this point. For the purposes of our exercise, we will use the default values offered.

To complete our project, only the third device is now required.

1. We then add a 1760-LDFC processor module, a 1760-DUB display module, and a 1760-IB12XOB4I0F I/O module to the Workbench using drag and drop.

2. A Pico-Link ID address is automatically proposed in the same way as when the second device was added. You can also determine the geographical location of the device. As we already have two devices on the Workbench, it is possible to place the third device between the first and the second at location 2.

In our case, we shall leave this field empty and accept the preset values by clicking OK.

Our training project is now complete so that we can continue with creating the program.

**Programming with Pico-Link Functions**

Our practice program needs to perform the following tasks:

- Pressing the cursor button P1 of device 2 ten times should set output Q1 on device 3 to “1”.
- Output Q1 of device 3 is to be reset to “0” via the push-button P2 on device 2.
- Counting is then to start from the beginning.

Device 3 is an input/output device, i.e. does not have its own circuit diagram.

We shall start with the program in device 2 (Pico-Link ID 2), which contains the reading of cursor buttons P1 and P2. Double-click the Project View on device 2 (Pico-Link ID 2) to switch to Circuit Diagram View.

As device 2 is a slave and cannot control the outputs of an input/output device, the signals have to be passed on to the master. This is implemented using the operand **SN bit output via the bus**. We use this operand to notify Pico-Link stations that P1 and P2 were actuated. Wire up your circuit diagram as shown in the figure below.
Program for Device 2 (Pico-Link ID 2)

The Properties fields for the two bit outputs must be set as follows:

1. To do this we return to Project View and left-click Device 2. The parameter dialog will appear in the Properties field.

2. Select the System tab and activate the P Buttons check box (see the following figure).

Enter the number of the operand in the SN list box, 1 for SN1 and 2 for SN2. The preceding station number (Pico-Link ID) must be set to 1 for both SN operands since the signals are to be sent to the Pico-Link station 1 (master).
Activating P Buttons

We have previously implemented the reading of the P buttons on device 2. We must now further process the information obtained, i.e. the pressing of the P1 button is incremented and the counter state is reset to zero using P2. We wish to assign the logical processing of the data in our example to the master. As we are now in the Project View we can double-click Device 1 (Pico-Link ID 1) to activate circuit diagram entry for the master.

Once in Circuit Diagram View, we can start directly with the reading of the data sent from Device 2. This is implemented using the RN operand (bit input via the bus). The operand 2RN1 reads the sent value from 1SN1 (Pico-Link ID 2) and passes it on to a counter function block C1. Similarly, we read the second value sent from Device 2 (Pico-Link ID 2). 2RN2 is used for resetting the counter function block C1 to zero.

In conclusion, we can say that 2RN1 reflects the status of button P1 and 2RN2 the status of P2 of device 2.

Program for the Master (Device 1, Pico-Link ID 1)

Reading of bit values via the bus.

The RN operands must be set as follows
The following table shows the relationship between the SN and RN operands:

<table>
<thead>
<tr>
<th>Send bit via 1SN1 from Pico-Link ID 2</th>
<th>Read sent bit via 2RN1 to Pico-Link ID 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Bit is meant for station 1 (Pico-Link ID 1)</td>
<td>2: Bit comes from station 2 (Pico-Link ID 2)</td>
</tr>
<tr>
<td>SN: The bit is sent</td>
<td>RN: The bit is read</td>
</tr>
<tr>
<td>1: Operand number (is between 1-32)</td>
<td>1: Operand number (is between 1-32). Must be identical to the number of the SN operand.</td>
</tr>
</tbody>
</table>

The counter function block C1 is assigned the following parameters:

To reset the function block in line 2 to zero, select the coil function RE (Reset).

Once the counter function block has reached the value 10, we wish to set output 1 on device 3 (Pico-Link ID 3) to “1”. The existing program must be extended by one line (see the following figure).
Complete circuit diagram of the master.

To address the output of the input/output module (Pico-Link ID 3), rather than the local output Q1 of the master, we have to assign the output Q1 with Pico-Link ID 3.

Properties of output 3Q1.

That's it! The programs have been written and just have to be downloaded to the devices.

You can use the list box in the standard toolbar to quickly change the Circuit Diagram View in this project from one device to another, switching, for example, between the circuit diagram of device 1 to device 3. The device change is carried out by selecting the appropriate Pico-Link ID in the list box.

**Pico-Link Configuration and Communication**

To download the programs we have created, we must first prepare the hardware. To do this, we need three Pico GFX-70 devices that we connect up using the Pico-Link cable. We must also provide the first and last device with the bus terminating resistor.
We must also connect the PC to the master that is always represented by the device shown on the far left (see following figure). For this we require the 1760-CBL-PC02 PC connection cable.

Three stations connected via Pico-Link.

The figure above shows the Pico-Link configuration consisting of three stations (1), two Pico-Link connection cables 1760-CBL-INT01 (2) and two bus terminating resistors 1760-TERM1 (3).

Before changing to Communication View, check whether the correct COM interface is set on your PC. Choose Communication, Interface to determine which interface is selected and, if necessary, select another one.

We are now read. We then select Communication View using the appropriate button in the Toolbox.

If the change to Communication View is excessively long and an error message is output, check/change the selected interface.

Choose Communication, Online to restart the connection establishment.

Once the Communication View has been called up, we must then configure the Pico-Link.
We now simply have to start the configuration via Communication, Configuration, Pico-Link. PicoSoft Pro starts to transfer the Pico-Link intraconnect configuration to the master immediately. Each Pico-Link station is assigned their own Pico-Link ID (in this case, 1 - 3), and the valid parameters for baud rate and bus delay. After the transfer has been completed, you can read the Pico-Link ID in the device display as NT1-NT3. The Pico-Link LED starts flashing to indicate that the Pico-Link is functioning correctly.

The programs can now be downloaded without any problem. To avoid having to download every program onto their respective devices individually, let us switch beforehand from local to Pico-Link communication (via the master). Choose Communication, Device, Pico-Link Operation to specify the device with which the PC is to be connected. In our case, this is Device 1.

Choose Communication, Program, Download to start the download of the individual circuit diagrams or all of them. In the selection dialog (see following Figure) select All so that no programs are forgotten.

Click OK to start the program transfer to the devices.

Program selection dialog in Master mode.
To start program execution, open the **Program** dialog in the Toolbox and click the **RUN** button. As **Remote RUN** was set on all devices, this command is applied to all three devices, i.e. all devices in the Pico-Link are switched from **STOP** to **RUN**.

As mentioned in previous lessons, we can display the states in the devices in PicoSoft Pro via **Communication, Status Display On**.

As we have selected **Device 1** via **Communication, Device, Pico-Link Operation**, we can view the online power flow in the master.

When pressing the P1 cursor button of the second device, you will see very clearly that the information is passed on successfully to the master, where the counter function block is incremented.

---

**Circuit diagram status display of the master program in device 1** (when pressing cursor button P1 of Device 2).

**Function block status display of the master program in device 1** when pressing cursor button P1 of Device 2.
1. Move from the circuit diagram to the function block diagram via View, Function Blocks or directly via the button.

2. Return to the Circuit Diagram View by clicking the button once more.

You may wonder how you can display the states of the second device in PicoSoft Pro. Do I have to disconnect the PC from the device and reconnect the 1760-CBL-PC02 somewhere else? No, it is much easier to do.

1. Choose Communication, Device, Pico-Link Operation to select Device 2. PicoSoft Pro will then fetch its data for display from Device number 2 (Pico-Link ID 2) and not from the master.

![Circuit diagram display of the slave program in device 2 (when pressing cursor button P1)]

2. After the cursor button P1 has been pressed ten times on Device 2, the Q1 output of the third device (Pico-Link 3) is set to logic “1”.

Congratulations!

You have now successfully worked through the PicoSoft Pro Beginners' Course. Hopefully, you have acquired some idea of how simple and versatile the Pico controllers are to use.
Rockwell Automation provides technical information on the web to assist you in using our products. At http://support.rockwellautomation.com, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration and troubleshooting, we offer TechConnect Support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://support.rockwellautomation.com.

Installation Assistance

If you experience a problem with a hardware module within the first 24 hours of installation, please review the information that's contained in this manual. You can also contact a special Customer Support number for initial help in getting your module up and running:

| United States | 1.440.646.3223  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Monday - Friday, 8am - 5pm EST</td>
</tr>
<tr>
<td>Outside United States</td>
<td>Please contact your local Rockwell Automation representative for any technical support issues.</td>
</tr>
</tbody>
</table>

New Product Satisfaction Return

We test all of our products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned:

<table>
<thead>
<tr>
<th>United States</th>
<th>Contact your distributor. You must provide a Customer Support case number (see phone number above to obtain one) to your distributor in order to complete the return process.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outside United States</td>
<td>Please contact your local Rockwell Automation representative for return procedure.</td>
</tr>
</tbody>
</table>

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1760-QS002A-EN-P - April 2004

Copyright © 2004 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.