

# **Unity Programming Software**

## **Telemecanique Modicon M340 Programming**



## Table of Contents

	Page
A. Starting a Project With the M340	Unity-3
1. Configuring the Processor	Unity-3
2. I/O Configuration	Unity-3
3. Starting Ladder Program	Unity-5
4. Processor Download	Unity-5
B. Ladder Logic Programming	Unity-6
1. Two Contacts in Series (Logical AND)	Unity-6
2. Entering Branches (Logical OR)	Unity-8
3. Timers	Unity-8
4. Counters	Unity-9
5. Online Editing With PLC in Run Mode	Unity-10
C. Adding Comments to Ladder Logic	Unity-10
D. Assigning Symbols to Addresses	Unity-10
E. Displaying Memory Locations	Unity-11
F. Input/Output Forcing (Disabling)	Unity-11
G. Importing	Unity-12
1. Importing Database from TXT file	Unity-12
2. Exporting Database to TXT file	Unity-12
3. Importing PLC Program	Unity-13
4. Exporting PLC Program	Unity-13
H. Saving Ladder Logic Program	Unity-14
I. Restoring Ladder Logic Program	Unity-14
J. Printing Ladder Logic Program	Unity-15

## Unity - 2

This document is a supplement to the “Lab 1.doc” lab exercise that introduces one to the basics of ladder logic programming. The following equipment is assumed:

- Schneider BMX XBP 0800 8-slot chassis
- Schneider BMX XBP 0800 8-slot chassis
- Schneider BMX CPS2000 power supply
- Schneider BMX P34 1000 M340 processor
- Schneider NOE 0110.2 Ethernet communication module in slot 1
- Schneider DDI 1602 16-channel 24-v discrete input in slot 2
- Schneider DDO 1602 16-channel 24-v discrete output in slot 3
- Schneider AMI 0410 4-channel analog input in slot 4
- Schneider AMO 0210 2-channel analog output slot 5

This document is primarily for Rev 4.0 of the Unity Pro software. A USB cable connection from the PC to the processor is also assumed.

In order to become familiar with basic ladder logic programming and addressing techniques on the Telemecanique Modicon M340 equipment, the steps to accomplish the first part of the first lab exercise are outlined in sections A, and B. The remainder of the sections outlines the steps to document your ladder, to print the ladder logic program, and to save your ladder logic diagram. The last sections explain the differences when programming offline.

In an industrial setting, the initial PLC programming is done offline and then later downloaded to the PLC when ready for the final test and debug. In addition, there is usually only one project (program) for each PLC on the network. For these labs, the ladder programming will be done online so that changes can be immediately tested.

Unity Pro S version 4.0 is used to program the M340 PLC's. An icon for this program should already be on the desktop.

The PLC and the PC used to program it are connected directly to the PLC either through a USB cable (the preferred connection) or through a serial port, called the Modbus protocol.

For the casual user, the Unity software is best operated using the point-and-click approach. Towards the top of the screen, buttons for frequently-used commands are displayed in toolbars. One may click on the button, or access the function by selecting commands from the menu at the top of the screen.

## A. STARTING A PROJECT WITH THE M340

Wire the switch/light board before proceeding. Power up the computer, the PLC rack, and external power supply, as needed.

If the Unity Programming Software main menu is not already displayed, click on the Unity icon (a pyramid). A blank main screen should appear with a list of selections on the menu across the top of the window. **If your project is already in the PLC**, open your project, select **PLC | Connect...** Open the ladder logic section and the first network should appear. Skip to the online programming functions.

When starting a project for the first time, it is best to initially set it up offline, and then go online. The initial setup consists of specifying the processor and configuring the discrete I/O modules.

### 1. Configuring the Processor

From the menu bar select **File | New...** or click on the paper sheet icon. A window will appear asking for the processor type. Select the line with BMX P34 1000 and click on **OK**. A project browser window should appear on the left side of the screen. On the line with “0: PLC Bus”, click on the “+” to expand it to show the backplane. The power supply and the processor (in slot 0) should appear. Double-click on the line with “0: BMX XBP 0800”. The project browser window will shrink to half-size and a hardware catalog window will appear below or above it. In addition, a pictorial representation of the modules in the backplane will appear (with only the power supply and processor).

Configure the serial port to be Modbus. On the line with “0: BMX P34 1000”, click on the “+” to expand it to show the serial port. Double-click on “SerialPort” and set the following options:

Type – Slave	Transmission Speed – 19200 bits/s
Slave number – 1	Data – RTU
Physical Line – RS232	Stop – 1 bit
	Parity – Even

Close the Serial Port configuration window. If a pop-up appears asking if you want to validate your modifications, click on **Yes**.

### 2. I/O Configuration

Now we need to populate the remainder of the backplane. The module in slot 1 is the Ethernet communication module, part number NOE 0110.2. Expand the Communication section of the hardware catalog. Click and drag the NOE 0110.2 to slot 1 (a dashed rectangle is dragged). Release the mouse button and the NOE 0110 module will appear in slot 1. In similar manner, place the following modules in the backplane

Slot	Module	
2	DDI 1602	16-channel 24-v discrete input
3	DDO 1602	16-channel 24-v discrete output
4	AMI 0410	4-channel analog input

5 AMO 0210 2-channel analog output

Note that the discrete modules are in the “Discrete” section of the catalog and the analog modules are in the “Analog” section of the catalog.

Close the hardware catalog by clicking on the small “x” in the upper right corner of the hardware catalog window **but do not close the program by clicking on the “X” in the upper right of the Unity window.**

Now add an Ethernet network, needed for configuring the Ethernet module. Expand the **Communication** section of the Project Browser window and do the following:

- Right-click on “Networks”
- Select “New Networks”
- In “List of Available Networks” field, select “Ethernet”
- Accept the default “Change Name”, should be “Ethernet\_1”
- Click on **OK**.

Now configure the Ethernet channel on the Ethernet communication module (NOE 0110):

- Right-click on the NOE module and select **Open Module**.
- Click on “Channel 0” in the left part of the window that appears.
- In the function field (lower left), select ETH TCP IP.
- In the Net Link field, select “Ethernet\_1” (the network you configured above).
- Now close the window by clicking on the “x” in the upper right corner of the module window. If a pop-up appears asking if you want to validate your modifications, click on **Yes**.

Now set up the IP address in the Ethernet module

In the Project tree, in **Communications**, double-click on “Ethernet\_1” network. A configuration window should open. Populate the fields with the following:

- Configured – selected
- IP address – whatever IT assigns
- Subnetwork mask – whatever IT assigns
- Gateway address – whatever IT assigns
- Ethernet II - selected

Close the Ethernet\_1 window by clicking on the “x” in upper right corner. If a pop-up appears asking if you want to validate your modifications, click on **Yes**.

Now assign symbols to the input channels. Double-click on the “Elementary Variables” in the Project Browser (part of the “**Variables & FB Instances**” section). A Data Editor window will appear. To assign symbols to the discrete channels,

- Click on the **Name** column in the (blank) last line.
- Type your symbol name, in this case “Start\_PB” and click on the **Type** column in that row.
- Type “EBOOL” or click on the down arrow next to the box in the Type column. Select EBOOL for Boolean (discrete) data type. Click on the **Address** column.
- Type in the proper input address for the Start\_PB input channel, which is “%I0.2.0”, representing chassis 0, slot 2, and channel 0.

Click on box in the **Comment** column and type in a description, for example, “Start push button switch”.

Repeat the above 5 steps to specify the symbols and descriptions for the other input and output channels. Addressed for output channels start with “%Q”.

You should note that when clicking on the **Type** column, it fills in the data type from the previous row.

When finished, close the Data Editor window by clicking on the “x” in the upper right corner below the “x” that closes the program or by clicking on the “x” of the Data Editor window if it is not full-size.

Now a blank ladder must be started.

### 3. Starting Ladder Logic Program

In the Project Browser, expand

**Program**

Tasks

MAST

Right-click on “Sections” (below MAST) and select *New Section....* A "New Program Section" window will appear. Click on the down arrow next to the Language field and select LD to select ladder logic. Type in a name for the section (no spaces, 32 characters maximum) and press **OK**. The blank ladder should appear as a window.

### 4. Processor Download

The process of transferring a project to the processor consists of the following steps:

Build project

Connect to processor

Transfer program

Before one can download the processor, the project must be “built.” A yellow indicator in the lower right of the window indicates that the program has been changed and must be built before downloading to the processor. First, one should use the “analyze” function to check for errors in the program. To analyze the program, select **Build | Analyze Project** or click on the icon with the three sheets of paper with an arrow pointing down. Status messages, including errors and warnings appear in the window at the bottom of the screen. If there are no errors, then proceed with “building” the project. To analyze the program, select **Build | Rebuild All Project** or click on the icon with the dots inside the rectangle and three arrows pointing down. Status messages, including errors and warnings appear in the window at the bottom of the screen. If successful, a green BUILT indicator should appear in the lower right.

#### **IMPORTANT:**

Check the connection to the processor.

From the main menu, select **PLC** Verify that the **Standard Mode** is selected (third line, icon is depressed).

Select **Set Address...**

Set the Address to "SYS" and the Media to "USB".

Click on the "Test Connection" button.

After a few seconds, a "Successfully connected to the target" message should appear.

If not, check cable connections

Click on "**OK**".

Click on "**OK**".

Connect to the processor. From the main menu, select **PLC**. Select **Connect...** Some indicators should appear at the bottom of the window. Namely, a red "DIFFERENT" box should appear if the program in the processor is not the same as what you are editing. This should appear anytime you are downloading your program for the first time during a lab session or if you make any program changes off line and then connect to the processor.

To download to the processor, select **PLC | Transfer Project to PLC...** A "Transfer Project to PLC" window will pop up. Information about the program being downloaded is on the left side and information about the program in the PLC is on the right. Check the "PLC Run on Transfer" box if you want the PLC to immediately start running after a successful transfer. Click on **Download**. If the controller is currently running, a window will appear, confirming that it is okay to stop the PLC. Click on **OK**. A window will appear showing the transfer progress. If you checked the "PLC Run on Transfer" box, a window will pop up confirming the change to run mode. Click on **OK**.

You are now on-line, indicated by the RUN green box or STOP yellow box at the bottom of the screen.

If error message pops up indicating that CPU compatibility check failed, you specified the wrong processor. To correct, right-click on the processor and select **Replace Processor**. Click **Yes** on the warning pop-up message. Select the correct processor and then click on **OK**. If a pop-up appears asking if you want to validate your modifications, click on **Yes**. Rebuild the project.

## **B. LADDER LOGIC PROGRAMMING**

The STOP yellow box should appear at the bottom of the project window. If not, click on the Stop icon, or select **PLC | Stop**. The steps necessary to program the first part of the exercise, the series rung, will be explained in detail. For the other parts of the exercise, only the different steps required to do these parts are explained.

The ladder logic window is marked off in rectangles, whose corners are indicated by the dots. Contact and coil instructions occupy one rectangle. Timer blocks occupy four (vertically stacked) rectangles. Counter blocks occupy five or seven rectangles. Math blocks occupy at least three rectangles, depending on the number of inputs. The position of the cursor is indicated by the gray box.

The ladder logic window background is grey if program is animated (monitoring a program). When editing (even if editing online), the background is white. When starting an online edit while animating (grey background), you will be asked to confirm the modification. If you click on **Yes**, the background will change to white (indicating that animation is disabled).

Also note that when editing online, the changes are not saved on the PC until you disconnect from the PLC. Then Unity will ask if you want to update Upload Information. Always click *Yes* in response to this prompt.

## 1. Two Contacts in Series (Logical AND)

The first ladder program to be implemented will be a simple series control rung. Two switches, SW3 and SW4, will be placed in series and will be used to control a single light, LA1.

To put in the contacts and coils, do the following:

If the ladder background is gray, click on the computer icon, or from the top menu, select **PLC | Disconnect**

A normally open (NO) relay contact is needed at the input, so click on the appropriate contact symbol (-| -) above the browser/ladder window.

Move the cursor arrow to the upper left rectangle and click the mouse. (If you are online and the background is grey, you will be prompted to confirm the modification. Click on *Yes*.) The NO contact will appear in the rectangle, connected to the left power rail.

Now move the cursor to the next rectangle to the right of the NO contact and click the mouse. A NO contact should appear, connected in series to the first contact.

Click on the output coil symbol (-() -) above the ladder window.

Move the cursor to the rectangle to the right of the second contact and click the mouse. An output coil should appear, connected in series to the rightmost contact.

Click on the arrow icon above the browser/ladder window to disable the instruction-entry mode of the software. Now the cursor can be used to select the elements on the screen, and will not place new instructions.

Double-click on the first contact and a window will popup, asking for the variable name for the contact.

Switch SW3 will be used first, so type the variable name for that switch (“SW3”) and press <Enter>. Alternatively, if you have forgotten the symbol and/or address, click on the button with the dots to display a list of defined variable names and select the appropriate one from the list. Also, if you forgot to define symbols, clicking on the Elementary Variables in the Project Browser window will bring up the variable editor.

Double-click on the other NO contact. Type the variable name for that switch (“SW4”) and press <Enter>.

Double-click on the output coil and type the variable name for light LA1 (“LA1”) and press <Enter>.

The right box along the bottom of the window should show “NOT BUILT”. In order to run this program, the project changes must be built and downloaded into the PLC. The changes you just made were not actually entered into the PLC memory. Click on the Select **Build | Build Changes** or select the Build Changes icon (dots inside the rectangle and **two** arrows pointing

down.). The right box along the bottom of the window should show “BUILT”. The ladder window background will change to grey indicating that you are viewing the online program.

Click on the Run icon to start the program running. The controller is now set to respond to input conditions and perform appropriate output commands. If the background is not gray, turn on animation, by selecting *Services* | *Animation*. The ladder window background will turn to gray and the lines should change to red. The left power rail changes to green.

Try the input switches in various combinations and observe the output. Also observe the changing status of ladder components on the screen. Energized or true conditions cause the elements to be highlighted (green) on the ladder diagram. Note that if SW3 is off, and SW4 is on, the second contact does not highlight, but the “SW4” changes to green to indicate that the switch is on, even though power is not flowing through the contact, due to SW3 being open.

## 2. Entering Branches (Logical OR)

It is not necessary to stop the processor to make online changes to the program. Unfortunately, adding parallel branches is cumbersome until one becomes familiar with it. To do the exercise for Lab 1, the SW4 contact is moved to the node below the contact for SW3. To place the SW4 contact in parallel with the SW3 contact,

Select the SW4 contact and move it beneath the SW3 contact (click and drag).  
Select the vertical link icon, "|". Place the cross hair slightly to the right of the SW3 contact and click the mouse. A vertical link should appear with black squares at both ends. Placing the vertical links is the trickiest part of working with Unity. It was actually harder in the previous version of Modicon programming software.  
Select the horizontal link icon (between the vertical link and double arrow icons).  
Click in the rectangle formerly occupied by SW4. The LA1 coil is now joined with the right side of the contacts.

If you stopped the processor, change it back to run mode. The controller is now set to respond to input conditions and perform appropriate output commands.

## 3. Timers

To insert a block into ladder logic, the Types Library Browser window must be displayed. To display it, either select *Tools* | *Types Library Browser* or click on the appropriate icon (hard to describe, it is towards the right side of the icon bar, to the right of the icons with the magic wands). Most, if not all, of the blocks will be in the library named “Base Lib”.

A timer occupies four (vertically stacked) rectangles on the screen. To place a timer block in the LD diagram:

Select the appropriate timer as the next function block instruction to be inserted.  
In the Types Library Browser window, make sure the “Base Lib” library is selected.  
Set the Family name to “Timers & Counters”

Select Timer in the group window.  
 Scroll down to the appropriate timer (TON, TOF, or TP).

To insert the function block instruction,

Click on the appropriate timer in the Types Library Browser window and drag it into the diagram. It should be in the third or greater column (it can be moved later, if needed). Release the mouse button.

The timer block will appear.

Deselect the timer block insertion by clicking on the arrow icon.

Double click on the timer box. A window will appear showing the input and output connections. Deselect the Show EN/ENO option. Click on **OK**.

Use the horizontal link edit tool to make connections to the IN input and to the Q output.

To enter the time interval as a duration literal, double click on the left side of the PT input wire. A text box will appear. Type in an appropriate time duration literal, for example, t#3s (3 seconds), t#2m15s (2 minutes, 15 seconds).

Press <Enter> or click on the green check mark.

When online, the value of the ET is shown in a yellow box attached to the ET output wire. Unfortunately, this box hides any variable attached to the ET output.

#### 4. Counters

Counter instructions are placed in the ladder rung in much the same way as timers. Remember to display the Types Library Browser window when adding counters.

A counter occupies five or seven (vertically stacked) rectangles on the screen. To place a counter block in the LD diagram:

Select the appropriate counter as the next function block instruction to be inserted.

In the Types Library Browser window, make sure the “Base Lib” library is selected.

Set the Family name to “Timers & Counters”

Select Counter in the group window.

Scroll down to the appropriate counter (CTU, CTD, or CTUD). It is not necessary to use the DINT, INT, UDINT, or UINT types.

To insert the function block instruction,

Click on the appropriate counter in the Types Library Browser window and drag it into the diagram. It should be in the third or greater column (it can be moved later, if needed). Release the mouse button.

The counter block will appear.

Deselect the counter block insertion by clicking on the arrow icon.

Double click on the counter box. A window will appear showing the input and output connections. Deselect the Show EN/ENO option. Click on **OK**.

Use the horizontal link edit tool to connect the CU and R inputs and the Q output.

To enter the preset value as a variable or literal, double click on the left side of the PV input wire. A text box will appear. Type in an appropriate integer value, or click on the box with the dots to look up an appropriate variable name. Press <Enter> or click on the green check mark.

When online, the value of the CV is shown in a yellow box attached to the CV output wire. Unfortunately, this box hides any variable attached to the CV output.

## 5. Online Editing With PLC in Run Mode

The ladder logic program may be edited while the PLC is in the Run mode. However, note that the actual edits are being made on the PC, not the PLC. The changes must be downloaded before the edits are executed. In many ways, the process of online editing is much simpler than the process for the Allen-Bradley PLC-5, SLC-500, or ControlLogix. However, remember that you cannot “back out” of your changes if they are incorrect. You must reenter the old ladder logic. If you ever need to make complicated on-line edits, you will appreciate the procedure of the A-B processors.

## C. ADDING COMMENTS TO LADDER PROGRAM

Select the comment icon (yellow rectangle with turned-down corner and “A” inside) and move the cursor to where you want to place the comment. Click on the mouse. A comment box will appear. Type in your comment. It can be moved to any position, as long as nothing else already occupies the position. To change the comment, double-click on the comment. To resize the comment block, drag the rectangles around the outside of the comment box.

## D ASSIGNING SYMBOLS TO ADDRESSES

For a large project, it is much easier to enter the symbols into an external file in the format of comma separated variables and import them into the database. However, for the purposes of this lab, it is probably easier to manipulate the database directly. There are times when one will want to assign a specific address to a symbol, or to correct an incorrectly entered symbol. Double-click on the “Elementary Variables” in the Project Browser (part of the “**Variables & FB Instances**” section). A Data Editor window will appear. The first column lists the variable (symbol) name, the second column lists the data type, the third column lists the address and the other columns show the initial value and the description.

To add a new variable/address, type in the new variable in the appropriate place on the blank (last) row.

To modify the variable name and/or description for an address, use the mouse or cursor keys to highlight the appropriate box for the address and type in the correct information. To move to another field, use the mouse to select another field, press <Enter> or use a cursor key.

To delete an address/symbol from the database, use the mouse or cursor keys to select the variable name in the record. Press the <Del> key. A confirmation popup appears. Click on **Yes**.

When finished, close the Data Editor window by clicking on the “x” in the upper right corner below the red “x” that closes the program or by clicking on the red “x” of the Data Editor window if it is not full-size.

## E. DISPLAYING MEMORY LOCATIONS

Use an Animation Table to build a table that shows the contents of multiple addresses (bit, integer, etc.) and allow one to modify them. This feature is very useful for debugging program operation and mathematical operations.

To **build** an Animation Table, do the following:

Right-click on **Animation Tables** in the Project Browser window and select “New Animation Table”.

A window pops up, allowing you to name the table. If you check the Temporary Table box, the table will not be saved when you close the project. When finished, click on **OK**.

A table will be displayed.

Type in variable names

Repeat the previous step for every variable name/address you want to monitor.

To **close** an animation table:

Clicking on the “x” in the upper right corner below the red “x” that closes the program or by clicking on the red “x” of the table window if it is not full-size.

To **open** an animation table:

Double-click on its name in the list below the **Animation Tables** in the Project Browser window.

To change the value of an internal variable in an animation table, first click on the **Modification** button (to make it look depressed). Now change the value. Click the Modification button again to disable modifications.

## F. INPUT/OUTPUT FORCING

For inputs, the force function allows one to override the actual status of external inputs by forcing individual discrete inputs to be ON or OFF. Input forces are written to the input image table and have no effect on the actual input. The value in the force table overrides the actual input. For example, if %I0.2.2 is forced to 1 (ON), it will appear in the input image and ladder logic as 1, regardless of the actual status of the device.

For outputs, the force function allows one to override the processor logic and force individual discrete outputs to be ON or OFF. Output forces are written directly to the output

device and do not appear in the output image table. The value in the force table overrides the value in the output image. For example, if %Q0.3.4 is forced to 1 (ON), the actual output will be ON, regardless of how it appears in the ladder and output image table.

Note: Enabling or disabling forces can result in sudden machine movement, possibly injuring personnel. Applying or removing forces while forces are enabled takes place immediately, with no prompt for confirmation. **USE EXTREME CAUTION WHEN USING FORCES!**

To force a particular input or output, the variable must be accessed through an Animation Table:

- Open an existing animation table or create a new one.

- A table will be displayed.

- Type in variable names to be disabled

- Click on the “Force” button (shows depressed) to enable the forcing function.

- Highlight the variable to be forced.

- Change the value in the “Value” column to the desired value by clicking on the appropriate icon in the middle of the top line of the AT window:

- Down arrow terminated in red line – force off.

- Up arrow terminated in green line – force on.

- Up arrow with a red “x” on it – delete force.

Note that when a variable is forced, its force value is preceded by an “F”.

Also note that the “Force” button must show as depressed to change any force state.

## G. IMPORTING/EXPORTING

### 1. Importing Database from TXT File

To import an address/tag/description database from a .TXT (tab delimited variable) file:

- Right-click on **Variables & FB Instances** in the Project Browser.

- Select **Import**.

- A pop up will ask to save current project. Click on **Yes**.

- An import window will open. Select files of type Data text file (\*.txt).

- Select file to import.

- Click on **Import**.

- If there are duplicates to variable names already existing, a window will pop up allowing you to cancel, keep all, replace all, or selectively keep, replace, or rename individual variables. The default selection is to rename all new variables.

- Generally, if you are redefining the variables for simulation purposes, select "Replace All".

- Click on **Ok**

## 2. Exporting Database to TXT File

To export an address/tag/description database to a .TXT (tab delimited variable) file:

Right-click on “Elementary Variables” in the Project Browser (part of the “**Variables & FB Instances**” section).

Select Export.

An export window will open. Select files of type Data text file (\*.txt).

Select folder for file to be placed. Name destination file.

Click on *Export*.

Note that this file uses tabs to delimit the entries. To use Microsoft Excel to open the file:

Open Excel

With Excel Open, select a text file type (\*.csv, \*.txt) and select the .txt file.

When the import assistant opens, select “Delimited” for the file. Click on Next.

In the next window, check the Tab option box as the column separator. Click on the Next button.

In the next window select in the display the column that contain the values of the variables (fifth column) and click on the Text option button for the column data format. Doing so keeps the same values as in the .TXT file. If this format is not used for the column, positive values will appear as negative when opening the .TXT file in Excel.

To save the file, save it with the Text (tab-delimited) option.

## 3. Importing PLC Program

To import a program section:

Right-click on "Section" in the project tree.

Select Import.

If there are duplicates to variable names already existing, a window will pop up allowing you to cancel, keep all, replace all, or selectively keep, replace, or rename individual variables. The default selection is to rename all new variables.

Usually, one will need to select "Keep All".

Click on *Ok*

## 4. Exporting PLC Program

To export a program section:

Right-click on the section.

Select Export.

An export window will open. You will have the choice of the export to include:

The contents of all DDTs (derived data types) used in the section

The contents of all DFBs (derived function blocks) used in the section

The contents of all the SR (subroutine) modules called by this section

It is probably safest to keep these selected. The type of the file depends on the language for the section.

Select folder for file to be placed. Name destination file.

Click on **Export**.

To export a task (MAST or FAST):

Right-click on the task.

Select Export

An export window will open. You will have the choice of the export to include:

The contents of all DDTs (derived data types) used in the section

The contents of all DFBs (derived function blocks) used in the section

It is probably best to keep these selected.

Select folder for file to be placed. Name destination file.

Click on **Export**.

## H. SAVING LADDER LOGIC PROGRAMS

Ladder logic programs (projects) may be saved for future use. By saving your ladder logic programs, you are able to resume your experiment, even though others have been using the same PLC. One usually saves the program to the default program directory on the hard disk and then saves the program to a flash drive for backup/transport to another computer.

If the PLC program does not match the program on the PC (but has the same name), first upload the program currently in the PLC,

If not already connected to the PLC, Connect to the PLC (**PLC | Connect**).

Select PLC | Transfer Project from PLC...

On transfer window, check the names and timestamps of the PC and PLC programs.

If they match, click on **Cancel**, no need to upload.

To upload, click on **Transfer**. The various sections of the PLC memory being uploaded will be indicated with messages.

To save the current project,

To save with the same project name, click on disk icon or select **File | Save**

To save with a different project name, select select **File | Save As**

To transfer a saved project, open Windows Explorer. Find the project file (extension .stu). Copy it and the \*.bak and \*.ztx files.

## I. RESTORING LADDER LOGIC PROGRAMS

You may restore your ladder logic program from the files on the computer hard disk to the PLC either before going on-line, or while on-line. Before going on-line, do the following steps:

Click on the Open File icon or select **File | Open**

The file selector window will appear, looking for file with \*.STU extension. Double-click on the appropriate project. The project is now loaded into the PC. Build your project.

Select **PLC | Connect....**

To download to the processor, select **PLC | Transfer Project to PLC...**

A “Transfer Project to PLC” window will pop up. Information about the program being downloaded is on the left side and information about the program in the PLC is on the right. Check the “PLC Run on Transfer” box if you want the PLC to immediately start running after a successful transfer. Click on **Download**.

If the controller is currently running, a window will appear, confirming that it is okay to stop the PLC. Click on **OK**.

A window will appear showing the transfer progress. If you checked the “PLC Run on Transfer” box, a window will pop up confirming the change to run mode. Click on **OK**.

## J. PRINTING A LADDER LOGIC PROGRAM

You may print your program only while disconnected from the PLC (off-line). In order for the print function to be active, something must be selected to be printed in the documentation configuration. The default is to have nothing selected.

To access the documentation configuration, double-click on **Documentation** in the Project Browser window. A window appears in the main project area of the screen. Click on the “tree” structural view option. Right-mouse the following parts and select **Include heading** for each:

- Variables & FB Instances
- Program
- Cross References

Right-click on the first line (“Project”) and set up the following fields:

- Project Name: Name of your project
- Variables & FB Instances – Sort by **Symbol**

You may print your program only while off-line and while the documentation window is displayed. To print documentation,

If the documentation window is not shown, double-click on **Documentation** in the Project Browser window.

Select (highlight) the top line.

Select **File / Print**. Verify that the correct printer is selected and then click on **OK**.