Essentials for Troubleshooting Ladder Logic _Level I

Student Manual

Presented By

Jim Joros

Essentials for Troubleshooting Ladder Logic Level II

This manual is provided to purchasers with the express knowledge and understanding that the author will not be responsible or anyway liable for any act done or omitted to be done on the basis of information contained in this manual, nor for any errors or omissions in the said information.

The examples and diagrams in this manual are included solely for illustrative purposes. Neither Rocky Mountain Technical Training LLC, nor the distributor, warranty or guarantee any of the examples described herein. Nor do we assume, and do expressly disclaim any obligation to obtain and include information other than that supplied herein.

The user is expressly warned to consider and adopt all safety precautions that might be indicated or implied by the activities described herein and to avoid all potential hazards. By following any instructions contained, the user willingly assumes all risks in connection with such instructions contained herein.

Specifically, Rocky Mountain Technical Training LLC, the developer and the distributor makes no representation or warranty that these programs are fit for any particular purpose. Rocky Mountain Technical Training LLC, the developer and the distributor shall not be liable for any special, consequential or exemplary damages resulting in whole or in part, from the user's use of or reliance upon this material. (In simple terms, this means if you suffer a loss through the use of the examples or anything associated with it, it's not our fault. In fact, in today's litigious society, we make no claims or guarantees for the examples given and by you using them you agree to hold us blameless for anything you may experience as a result of such use.)

To continue providing you world class training, your comments and suggestions are welcomed and appreciated.

What people are saying….

Robert McDougal
Maintenance Electrician at Ashgrove Cement Co

I took Jim's course on Allen Bradley PLC's about a year ago and I would say that I learned more from Jim's 4 day course than I learned in a whole quarter at Bellingham Tech school and North Seattle Community College.

I like how Jim gives the students time to absorb what they are learning. Jim starts from the very basics with handouts, a detailed power point throughout the course and a hands on practice with your own equipment.

I was amazed at how much Jim covered during that 4 day course. After all these years of classes and long quarters if finally stuck. Thank you Jim Joros! I highly recommend any course that Jim would teach on Industrial Electrical Equipment.

Frederick Rackow B.Sc.
Technologist II at City of Regina

Jim was a great teacher for this device. He is quite comfortable knowing the parts and features of the PLC and how to access them through programming. I found his instruction clear and concise. He brings a lot of field knowledge into his classes which helps keep the interest level high during the class.

Are you a responder...

or a bystander?

"True heroism is remarkably sober, very undramatic. It is not the urge to surpass all others at whatever the cost, but the urge to serve others at whatever the cost" -Arthur Ash

A hero is one who has **RESPONSE ABILITY**. Whenever you are called upon to troubleshoot equipment, you are expected to demonstrate the ability to respond and solve the problem. Responding to a situation but lacking the ability to solve the problem makes a person a bystander…an observer at best!

Troubleshooting PLCs begins with understanding that everything is event related .Example: Event 3 (energize fill solenoid) will take place only if Event 1 (Conveyor is not running) and Event 2 (Proximity Switch shows that box is in place) are true. The PLC scan cycle is to control the events of the process. Knowing the events or steps of the process gives you the edge for troubleshooting with **RESPONSE ABILITY.**

Index

Essentials for Troubleshooting PLCs

The History of PLCs

Before there were PLCs, there were relays. The problem with relays was:

1) Size
2) Speed was very slow: 20-60 ms.
3) Re-wiring was required if changes had to be made.
4) Plant electricians had to understand and know how to read electrical schematics in order to troubleshoot.

The Digital Revolution

In the mid 60's, logic gates were introduced to replace relays. Maintenance electricians had to learn Boolean Algebra to be able to read the logic gate diagrams and troubleshoot the systems. Most companies stayed with the relays until Hydra-matic, a division of General motors, in 1968 requested a device that would meet the following:

1) The controller would have to be a solid state device that could be placed on the factory floor.
2) It had to operate in an industrial environment with all its dirt, moisture, electromagnetism and vibration.
3) It had to be modular in form to allow for easy exchange of components and expandability.
4) It would have to be programmed and maintained in the form of relay logic-a method that plant electricians, engineers and technicians were already familiar with.

The first programmable controller.

MODICON (Modular Digital Control) became the first programmable controller. In 1970, the first PLC was installed at General Motors Oldsmobile plant and at the Landis Company in Landis, Pennsylvania. Dick Morley is credited as the father of the first PLC but Allen-Bradley copyrighted PLC.

What is a PLC?

A PLC can best be described as a computer that takes the place of wires and relays used in controlling industrial processing and manufacturing. It is a dedicated processor that repeats the same program over and over. The Outputs (motors, solenoids, lights, etc.) are **controlled** based on the conditions of the Inputs (switches, push buttons, etc.) and the **Program Logic** it has been taught.

Types of PLCs.

There are two basic types: Fixed and Modular.

Modular PLCs consist of a rack that can accept many different types of I/O modules that can simply slide into the rack and plug in.

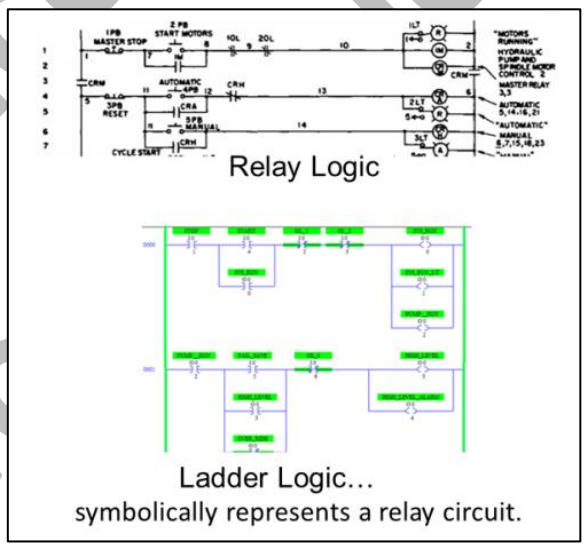
Fixed PLCs, aka “Bricks” due there size and shape, are usually smaller and the number of Inputs/ Output points are predetermined. The power supply is built within the unit.

Sizes of PLCs.

Modular PLC’s can be classified as small, medium or large depending on I/O points and memory sizes available,

PLC programming Language

Ladder Logic is a “program language” that symbolically represents a relay circuit. The outside rails of the ladder represent the power rails in an electrical schematic. The rungs in the ladder logic represent the lines in the schematic… often referred to as a line diagram. Symbols are also used to represent relays, N.O., N.C. contacts as well as timers, counters associated in relay logic. Ladder Logic is all virtual.



The challenge of troubleshooting PLCs …

The challenge for today’s technicians troubleshooting PLCs is the same as it was 45 years ago – to realize that although there are similarities between relay logic and ladder logic, troubleshooting PLCs cannot be approached with the same mindset that is used for relay logic. Failing to make a clear distinction between the two will only add confusion and frustration. Presently, the PLC industry is and has been attempting to move away from relay ladder logic programming, but it is finding it difficult to do so as a result of this early marketing strategy.

The analogy could be like saying; if you speak Spanish fluently then you should be able to speak Italian. Speaking in Spanish will only give you the chance to identify and understand some words, and rarely whole sentences, but does not mean you can go to Italy and speak and understand the language.

La analogía sería como decir; Si usted habla español con fluidez entonces podrás hablar italiano. Hablando en Español sólo le dará la oportunidad de identificar y entender algunas palabras y raramente toda condena, pero no quiere decir que puede ir a Italia y hablar y entender el lenguaje

L'analogia potrebbe essere come dire; Se parli correntemente spagnolo dovrebbe essere in grado di parlare italiano. Parlando in spagnolo solo vi darà la possibilità di identificare e comprendere alcune parole e raramente intere frasi, ma vuol dire non si può andare in Italia e capire e parlare il linguaggio.

Parts of a PLC

A PLC is made up of The following components:

- Power Supply/rack
- CPU
- Input and Output modules

1) Power supply/rack
   The power supply is available in 110 VAC, 220 VAC, 24 VAC or 24 VDC with regulated output voltages of 5VDC and 24 VDC. Racks sizes are available with 4, 7, 10 or 13 slots.

2) CPU (Central Processing Unit)
   Governs the systems activities including interpretation and execution of programmed instructions.

RSLogix 500 Processors Family.

SLC 5/01 This processor offers a basic set of 51 instructions with the choice of 1K or 4K of memory in a modular hardware configuration. Modular I/O systems that include an SLC 5/01 processor can be configured with a maximum of three chassis (30 total slots) and from 4 I/O points to a maximum of 3940 I/O points.

RSLogix 500 Processors Family…con't.

SLC 5/02 This processor offers additional complex instructions, enhanced communications, faster scan times than the SLC 5/01, and extensive diagnostics that allow it to function in more complex applications.

SLC 5/03 This processor provides 8 K, 16 K, or 32 K of memory. A built-in RS-232 channel gives you the flexibility to connect to external intelligent devices without the need for additional modules.

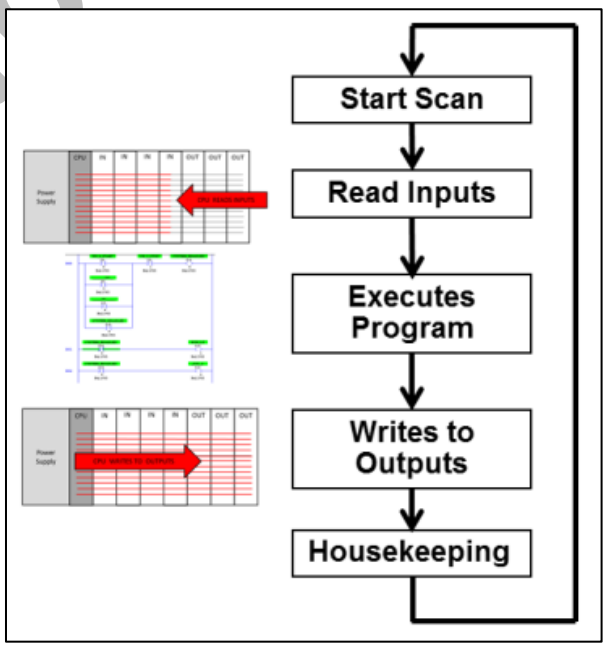
SLC 5/04 The standard DH-485 port has been replaced with a DH+ port, providing high-speed SLC 5/04-to-SLC 5/04 communications and direct connection to PLC-5 controllers.

The available memory options are 16 K, 32 K, or 64 K. In addition, there is a SLC 5/04P option, which is designed specifically for the Plastics Industry and contains ERC2 algorithms for Plastics Machinery Control.

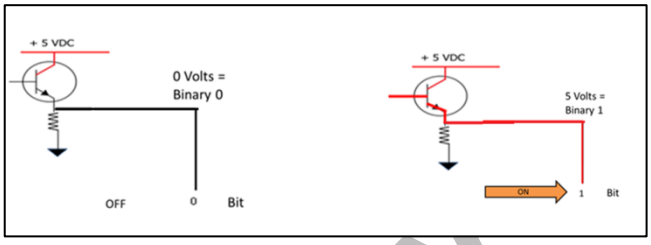
SLC 5/05 The SLC 5/05 processor provides the same functionality as the SLC 5/04 processor with standard Ethernet communications rather than DH+ communications. Ethernet communication occurs at 10 Mbps or 100 Mbps, providing a high performance network for program upload/download, online editing, and peer-to-peer messaging.

PLC Scan Time.

The PLC is a redundant processor that reads and executes the program over and over. The time it takes to go from beginning to end to scan the program is called the scan time. Scan times can be 1 or 2 ms.

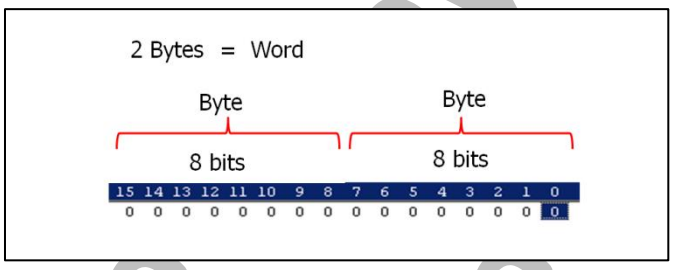


Bits – Bytes - Words



The smallest operation a processor can perform is turning a Bit ON or OFF

Bit stands for **B**inary dig**IT**

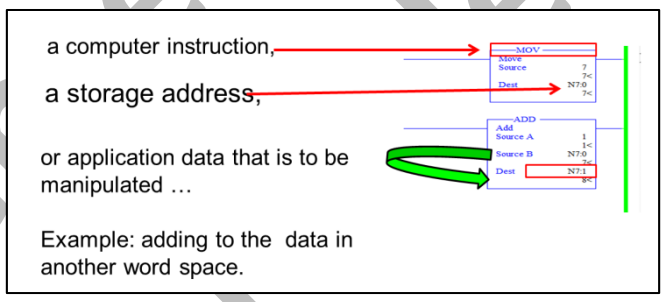


Eight Bits = a Bytes.

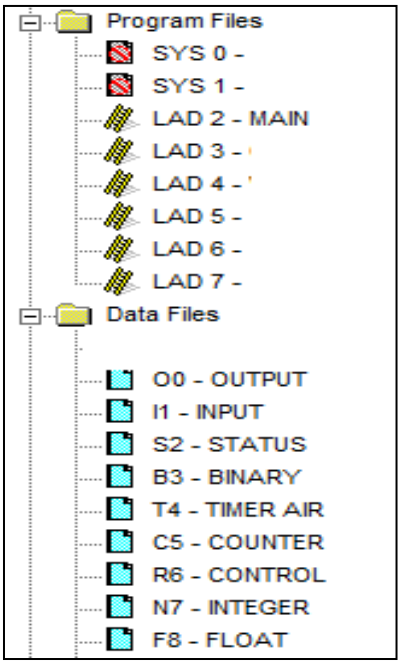
Two Bytes = a Word

A Word can be:

- ✓ a computer Instruction.
- ✓ A storage address.
- ✓ An application that contains data that is to be manipulated.



The memory organization is made up of two main categories: Program Files and Data Files.



Program Files are where ladder logic programs are stored.

System Functions [file 0]
This file contains various system-related information …Processor Type, I/O or file configuration, processor file name and password.

Reserved [file 1]
This file is NOT available to the user

Main Ladder Program [file 2]
Always included and contains user-program that defines how the controller is to operate.

Sub-routine ladder program [files 3 – 255]
… created by the user and are activated according to the sub-routine instructions that resides in the MAIN program.

Data Files…Stores information required to execute the program.

OUTPUT [file 0]
…stores the state of the output terminals for the controller.

INPUT [file 1]
…stores the state of the input terminals for the controller

STATUS [file 2]
…stores the controller information. This file is helpful in troubleshooting controller & program operation

BIT [file 3]
…used for internal relay logic instructions

TIMER [file 4]
…stores the timer's accumulated and preset values and the status bits.

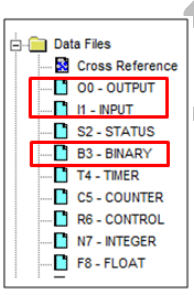
COUNTER [file 5]
…stores the counter's accumulated and preset values and the status bits.

Control [file 6]
…stores the length, pointer position and the status bit for specific instructions such as shift registers, sequencers etc.

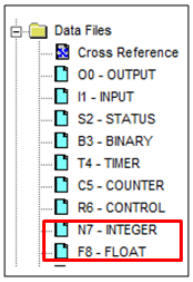
INTERGER [file 7]
…used to store the numerical values or bit information.

Floating Point [file 8]
…can store values in the range from + / - 1.1754944e to + / - 3.4028237e
Commonly used for 4-20ma inputs or outputs.

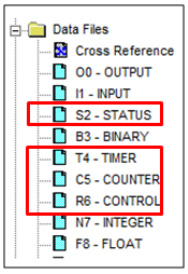
Data Categories All programs are made up of instructions using the Data Files



Status Bits



Words



Words and Status Bits

CPU Memory

ROM… READ Only Memory

- Computer memory that never changes.
- ROM memory contains data that is permanently recorded on the ROM chip.
- ROM is memory that is normally never erased or altered.
- ROM retains its data even when the power is off or disconnected.
- <u>It is for reading only</u>.

RAM *(Random Access Memory)*

- The CPU writes to and read from RAM
- The working memory of a processor.
- It stores input data, intermediate results, programs, and other information temporarily

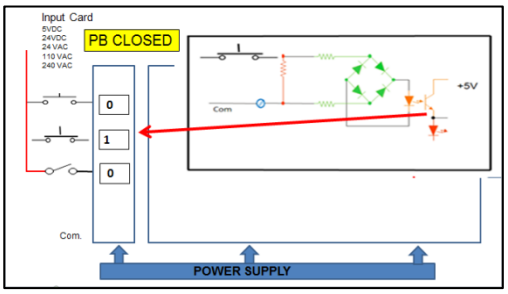
- Stores Variables:
  - Timer values
  - Counter values
  - Numbers, etc.

  Once the processor loses power, memory is lost. Battery backup provides protection for the data tables values.

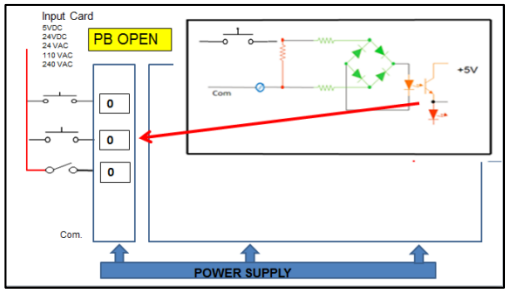
EEPROM … stands for Electrically Erasable Programmable Read-Only Memory and is a type of non-volatile memory. The EEPROM is a backup for the ROM. If changes are made to the program they must be loaded to the EEPROM. If changes are made to the program they must be downloaded and saved.

Functionality of Input modules.

Input voltages must be isolated and conditioned to a + 5 VDC level. (0VDC=0- 5VDC =1)

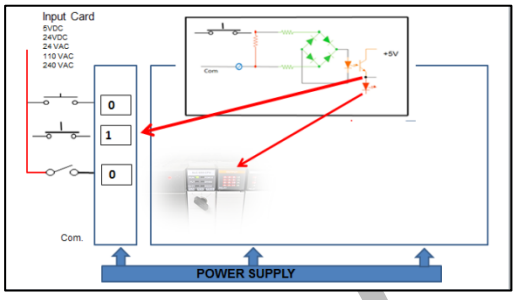


When a voltage is present at the terminal, it is conditioned to 5VDC representing a binary 1.(1 is ON– 0 is OFF)

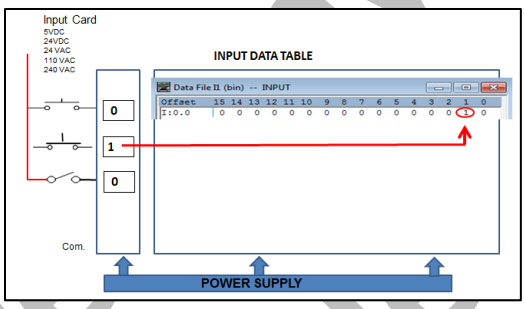


The front of the module has a LED that reflects the status of the terminal's screw voltage.

A 1 in the BIT BOX reflects 5VDC present at the terminal screw



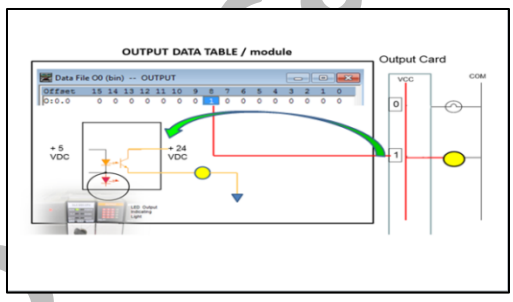
Each scan cycle, the CPU READS each INPUT and transfers the status to the INPUT DATA TABLE. (1,500 or more per second)



Functionality of Output modules.

Outputs are **WRITTEN** to and updates the outputs each scan cycle.

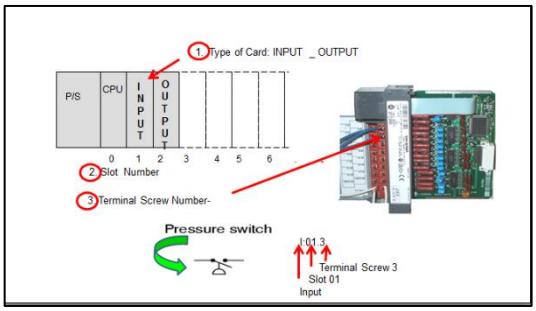
OUTPUT voltage from CPU (+5 VDC) must be isolated from OUTPUT VOLTAGES.



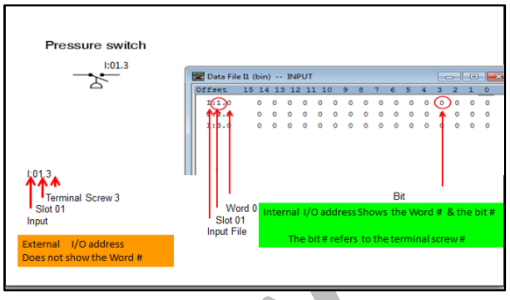
CPU **WRITES** the status of the OUTPUT DATA TABLE to the OUTPUT MODULE every scan cycle.

Parts of an I/O address.

1) Type of module Input or Output
2) Slot number
3) Terminal Screw #



Internal I/O address Shows: Type of module-slot location - Word # & the bit #



The terminal screw # refers to the bit #.

External I/O address does not show the Word #

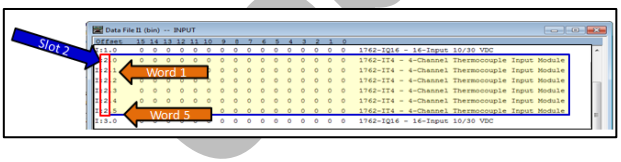
Internal Address shows::

Type of module Input or Output

Slot Location

Word

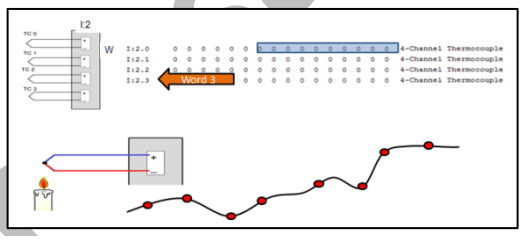
Bit



When a SLOT has more than 1 word …

Indicates:

- Remote I/O Addressing
- Specialty Card i:e Analog Module



Analog Module

An analog input card converts a voltage or current (e.g. a signal that can be anywhere from 4 to 20mA) into a digitally equivalent number that can be understood by the CPU. Examples of analog devices are pressure transducers, flow meters and thermocouples for temperature readings.

An analog output card will convert a digital number sent by the CPU to a real world output analog signal either as a voltage or current. Typical outputs signals can range from 0-10 VDC or 4-20mA and are used to drive VFDs, controllers, etc.

Analog modules can have 4 -8 channels and are also available with combinations of both Input and Output channels.

PLCs Numbering Systems

1) Decimal System
   In a base 10 numbering system, each digit in a position can have an integer value ranging from 0 to 9 (10 possibilities). The places or positions of the numbers are based on the powers of ten (e.g., units, tens, hundreds, thousands).

2) Binary
Binary uses only two symbols: typically 0 (zero) and 1 (one). The base-2 numeral system is a positional notation with a radix of 2. Each digit is referred to as a bit.

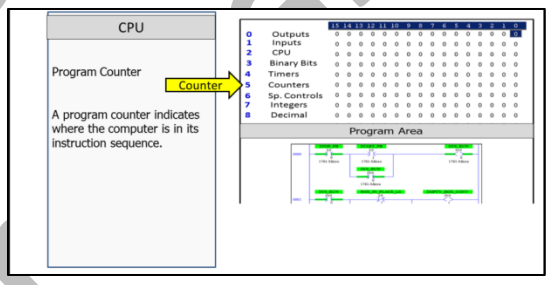
3) Octal
The octal numeral system is the base-8 number system, and uses the digits 0 to 7 with a radix of 8.

4) Hexadecimal
A hexadecimal is a positional numeral system with a radix, or base, of **16**. It uses **sixteen** distinct symbols, most often the symbols **0**–**9** to represent values zero to **nine**, and A, B, C, D, E, F to represent values **ten** to **fifteen**

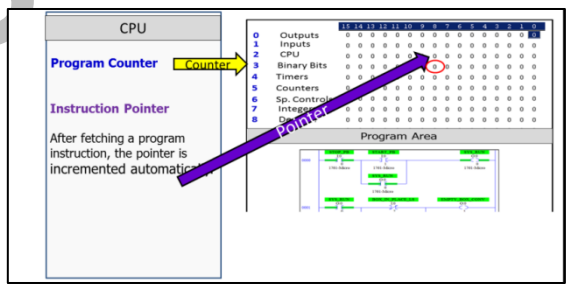
5) BCD
A binary code made up of 4 bits, that can represents 0-9 in the decimal numbering system.

CPU Counter and Pointer

A program counter indicates where the computer is in its instruction sequence.

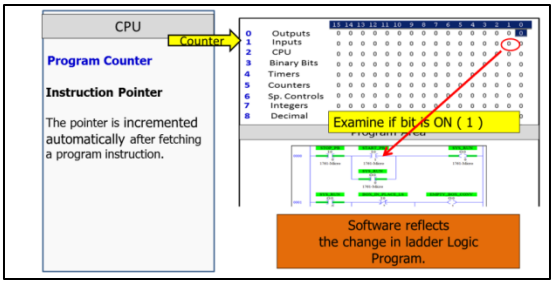


The Program counter holds either the address of the instruction being executed, or the address of the next instruction to be executed.



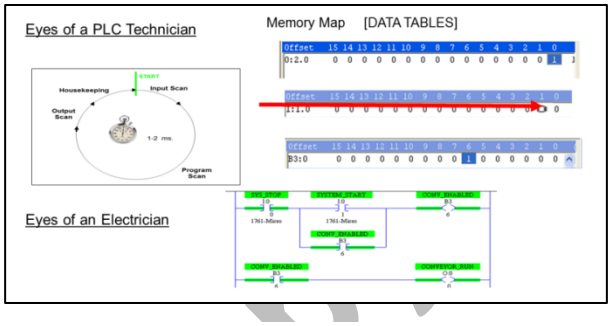
Instruction Pointer
After fetching a program instruction, the pointer is incremented automatically.

Software reflects the change in ladder Logic Program.



Think like a Processor

When looking at the program through the eyes of an electrician using relay logic, the program appears as being static.



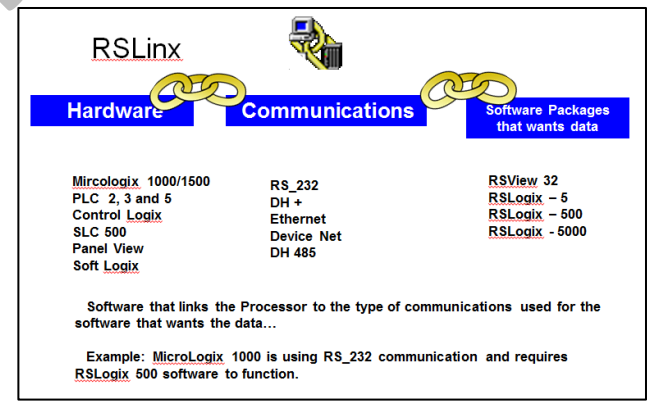
When viewing as a PLC technician, he/she must be aware that the program scan may be as fast as 1500 scans a second. The software that paints the symbols green resides in the PC and does not update as fast as the processor. When troubleshooting one needs to remember that if the color of the symbol does not appear to change state, it does not mean it hasn't taken place.

Software

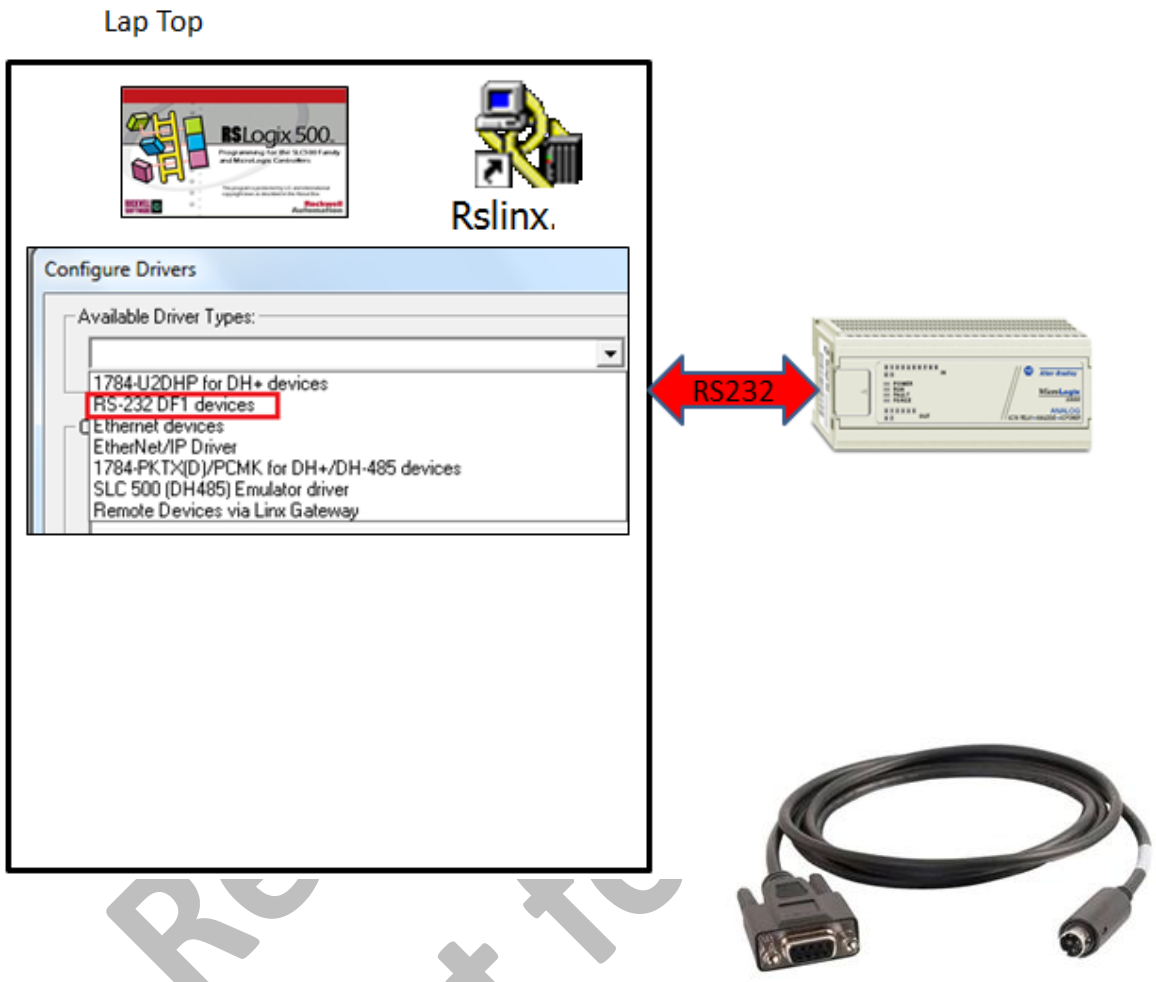
RSLogix500 performs all configurations and programming for…

all controllers
I/O modules
Communication modules and
Specialty modules

RSLinxs can be considered as a driver manager. The software links the hardware to the software packages that wants the data via the communication cables.



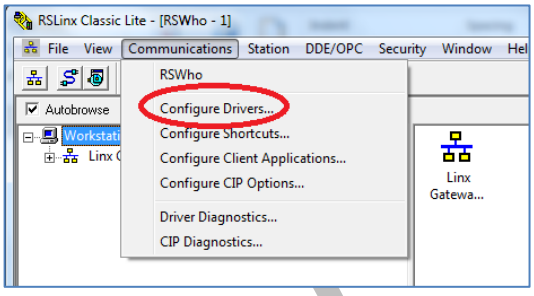
Configuring the “RS232” Driver using the serial communication port .



Configuring the “RS232” Driver using the serial communication port.

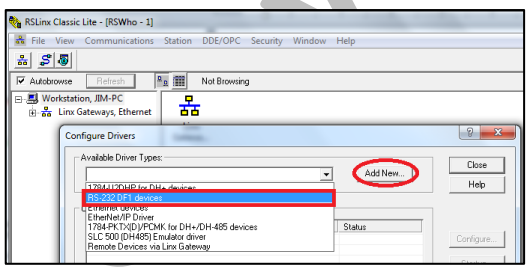
Open RSLinxs

Under “Communications” pull down menu, Click on “Configure Drivers”



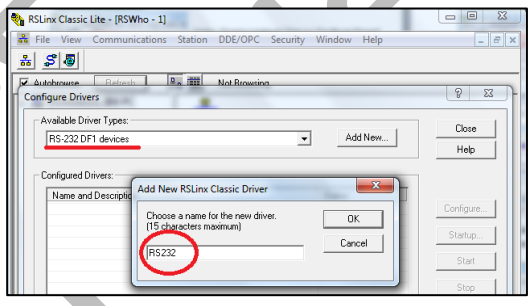
Pull down “Available Driver Types” and select “RS-232 DF1 devices”.

Click on “Add New”



Re-name “AB-DF1-1” to “RS232” .

Click OK

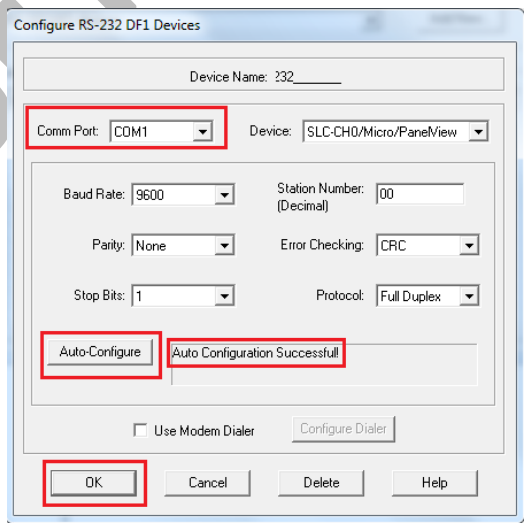


Select Com 1

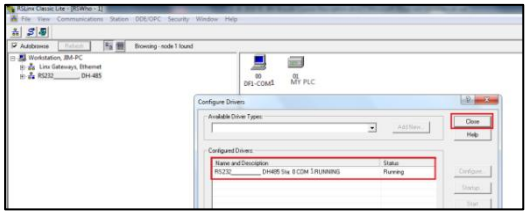
Click on “Auto Configure”

The software will validate the Com Port, cables and software and will display “Auto Configure Successful”.

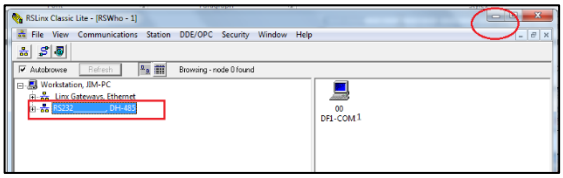
Click OK



Click on Close



Select the Driver on the left, minimize the Window and leave running in RSLogix .

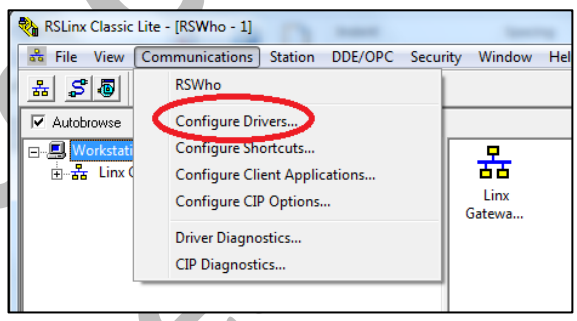


Configuring the “RS232 Driver” using a USB port

RS232 requires a 9300-USBS adapter

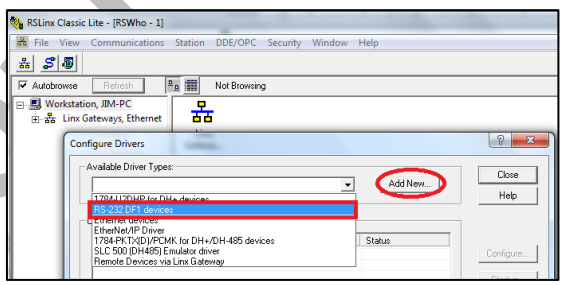
Open RSLinxs

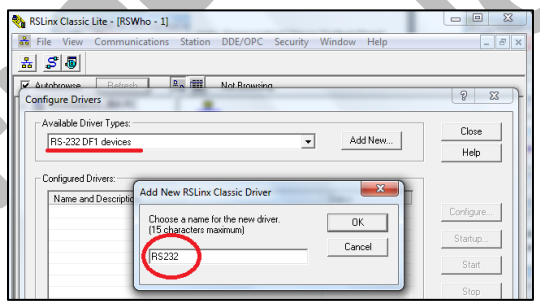
Under “Communications” Click on “Configure Drivers”



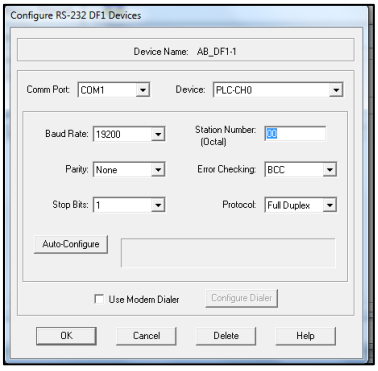
Pull down “Available Driver Types” and select “RS-232 DF1 devices”.

Click on “Add New”





Re-name “AB-DF1-1” to “RS232” . Click OK



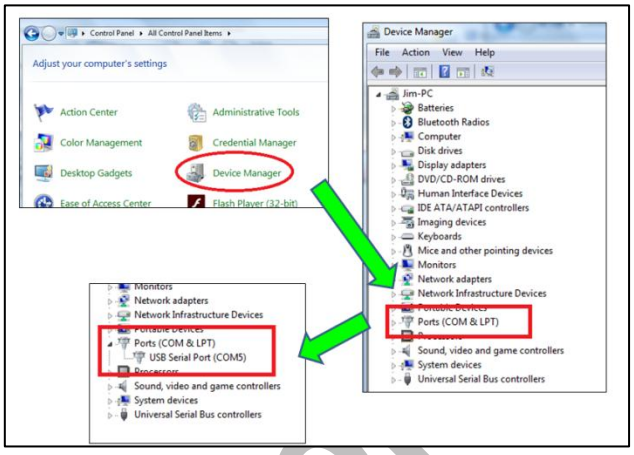
Identify the Com Port

Open “Control Panel”

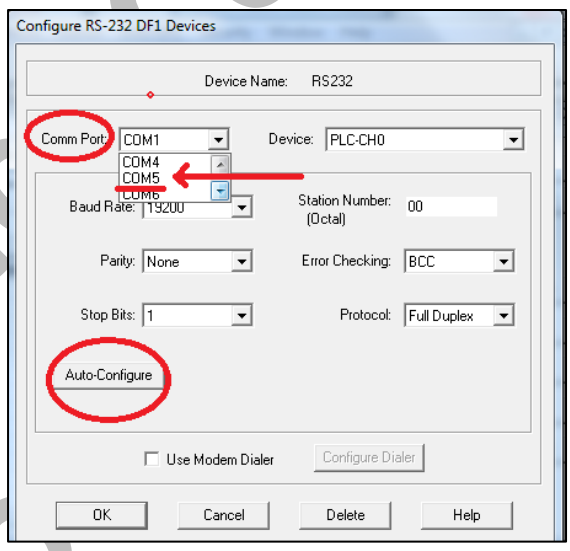
Open “Device Manager”

Open “Ports (COM & LPT)

Note Serial Com Port number assigned for PLC RS232 connection. In this example the com port is 5.



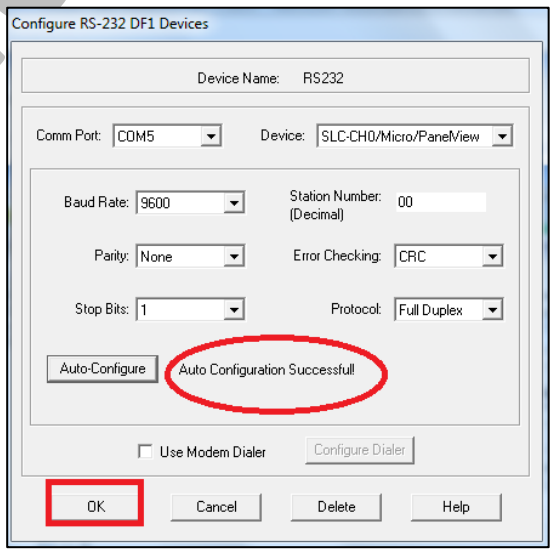
Go back to the Com Port pull down menu, select the Com Port number that was identified in the Device Manager and click.



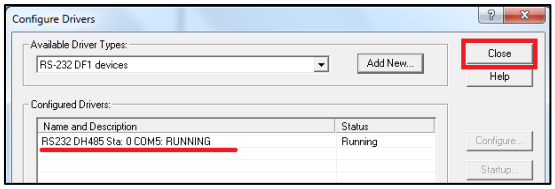
Go to “Auto Configure” and click.

The software will validate the Com Port, cables and software and will display “Auto Configure Successful”.

Click OK

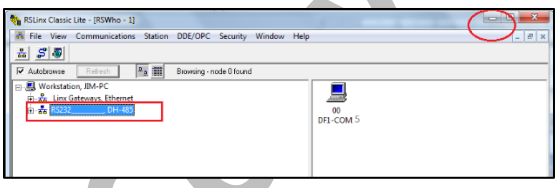


Click on Close



Select the Driver on the left, and select the PLC on the Right.

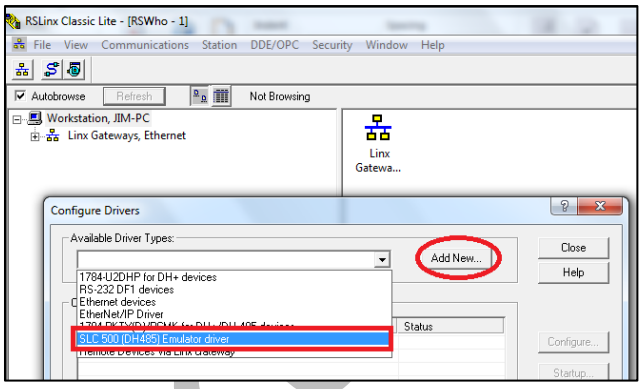
Minimize the Window and leave running in RSLogix .



Configuring the “Emulate 500 Driver”

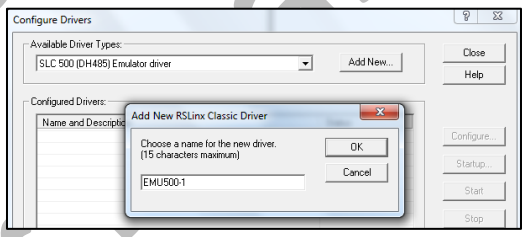
Open RSLinxs

Under “Communications”
Click on “Configure Drivers”



In the pull down menu “Available Driver Types”, select “SLC 500 (DH 485) Emulator Driver

Click on “Add New”



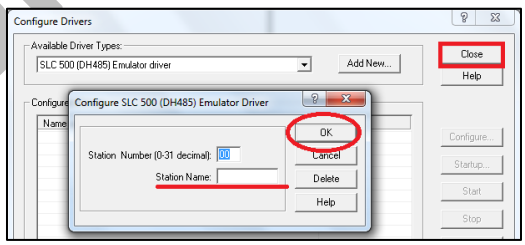
Leave name as default

Click OK

Add Station Name (Optional)

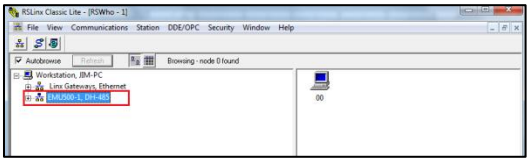
Click OK

Click Close



The Left Panel will display the EMU 500 driver with the network ICON being animated.

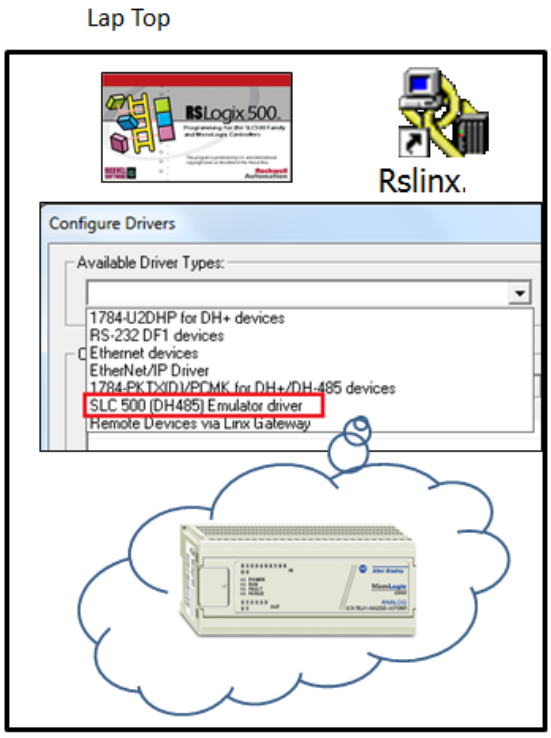
Minimize the window to allow RSLinx to run in the background.



RSLogix Emulate is a troubleshooting and debugging tool that can emulate most operations of Allen-Bradley PLC-5 and SLC-500 family processors. It executes the ladder logic programs in the computer, updating the programs' data tables, allowing you to approximate what is going to happen
when you download your programs to the physical processors.

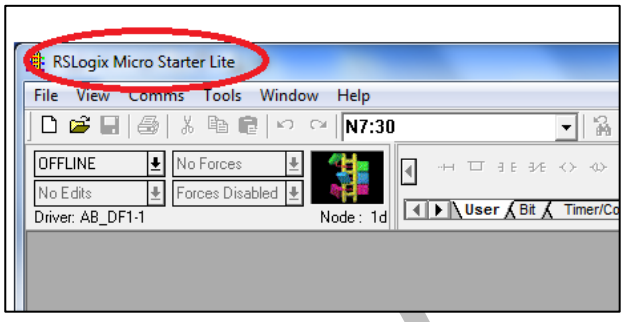
Since there is no real I/O, the emulated ladder logic takes cues only from the state of the data table. To generate responses in your ladder program, you will
need to change the value of the desired I/O bits, storage bits, or storage words acting as inputs to your ladder program.

The emulator scan cycle is approximately 15 – 60 scans per second vs. approximately 1500 scans per second to an actual processor. This slower scan cycle often allows you to observe the changing status of a bit which is helpful when learning how the processor functions,



Commission a new processor using the SLC 500 Emulator.

Open RSLogix Micro Starter Lite



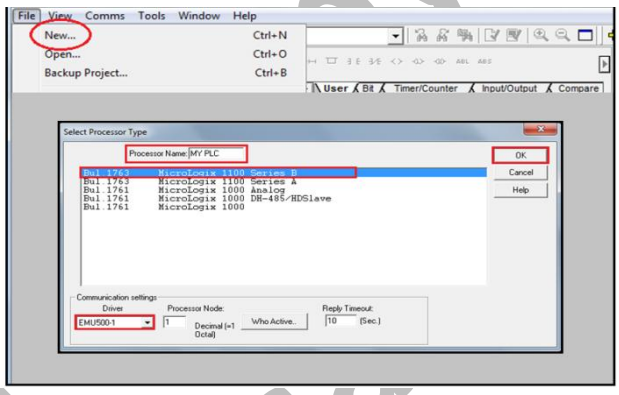
In the “File” pull down tab, select “New”

Change the processor name to “MY_PLC”

Select the MicroLogix 1100 processor.

Select EMU500 in Driver Configuration setting.

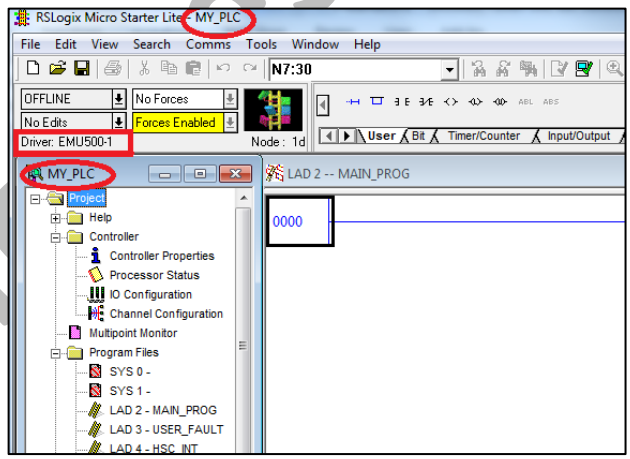
Click OK



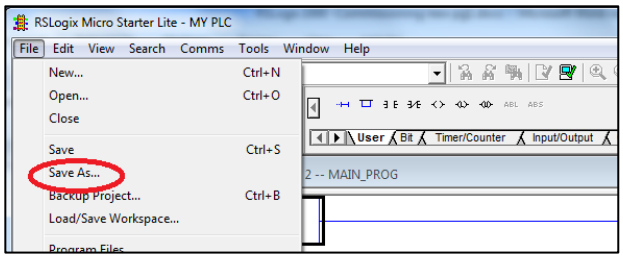
Processor Name

Name of Driver

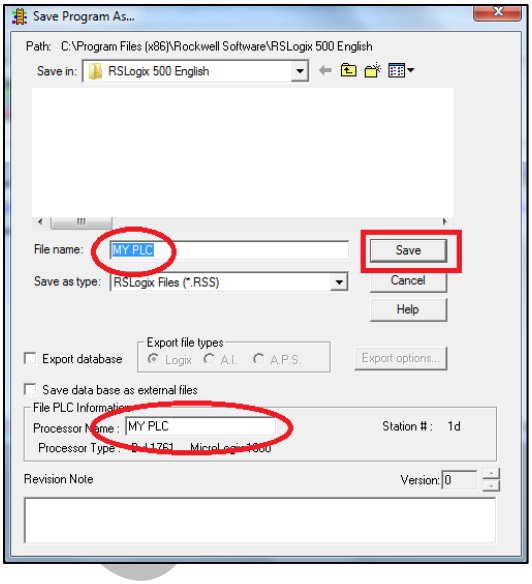
Project File Name



In the File pulldown menu Click on “Save As…”



Note that the PLC file name and the Processor Name are both the same. These names can always be changed in the “Save As…” window.

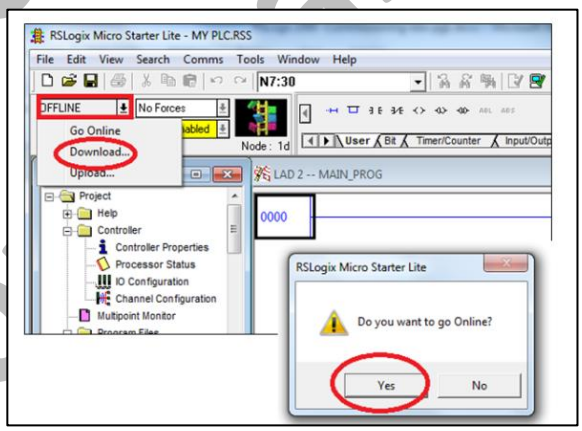


Click “Save”.

Compile

The program must be turn from a high level language into a machine level language or code, that the processor understands.
To compile you will need to download the program to the processor.

In the Offline pull down menu, Select “Download”.
Select “YES” when asked if you want to “Go On Line”.



Observe that the Ladder icon is animated indicating that communication has been established