Programmable Logic Controllers

Programming the Unitronics
V1040 PLC: Getting Started, Basic
Ladder Logic, and HMI Operation

VisiLogic Overview

- VisiLogic is the software tool you use to create control projects for Vision controllers.
  - Used to write, debug, and download the PLC control and HMI applications into the controller.
  - The PLC application is your control, or automation application
    - You write the PLC application using the Ladder Editor
  - The HMI application configures the operating panel's function
    - You use the HMI Editor to create the Displays that are shown on the controller's screen.
HMI Displays

• Displays tell your operators what to do
  – Provides status information about the system
  – Links to ladder logic being solved by the PLC
• You can have your operators enter data, and have the display instruct the operator what to do in case of a system problem or alarm
• A Display can contain both text and images
  – Text and images can be both fixed and/or variable

HMI Variables

• Variables are inserted into a Display to:
  – Show run-time values as integers
  – Represent run-time values with either text, images, or bar graphs
  – Show text messages that vary according to runtime conditions
  – Enable an operator to enter data using the Vision's alphanumeric keypad
VisiLogic Editors

- You use different editors to create your control project:
  - Hardware Configuration
  - Ladder
  - HMI Display
  - Variable
- Hardware Configuration
  - Define PLC type, communication, I/O modules, etc.
  - Similar concept to Allen Bradley hardware configuration
- Ladder Editor
  - Used to create the Ladder diagram that comprises your control application
  - Ladder diagrams are composed of contacts, coils, and function block elements arranged in nets

VisiLogic Software Layout
Hardware Configuration

• Select V1040 PLC type

Hardware Configuration

• Use "Connection -> Communication & OS" to define connection between PC and PLC.
• PC Port may vary
Ladder Editor

- In a Ladder diagram, the contacts represent input conditions
  - They lead power from the left Ladder rail to the right rail
  - This is why the first element in a net must always touch the left rail
- Coils represent output instructions. In order for output coils to be activated, the logical state of the contacts must allow the power to flow through the net to the coil
  - This is why the elements in a net must be connected
  - Each net must contain only one rung

Ladder Editor

- Use the Ladder Editor to:
  - Place and connect Ladder Elements
  - Apply Compare, Math, Logic, Clock, Store, and Vector functions
  - Insert Function Blocks (FBs) into your program
  - Build program Modules and Subroutines, and use internal Subroutine Jumps and Labels
  - Place Comments on Ladder nets
- Ladder elements and functions may be dragged and dropped between nets
Basic Ladder Logic Example

Operand Types

- We will use the following operand types in our initial examples
  - MB – Memory Bit
  - MI – Memory Integer (16-bit)
  - ML – Memory Long (32-bit)
  - T – Timer (32-bit)
  - C – Counter (16-bit)

- Other operand types will be introduced as needed
Basic Ladder Elements - Contacts

- Direct Contact -- a normally open (NO) contact
- Inverted Contact -- a normally closed (NC) contact
- Positive Transition Contact (Rise)
  - A one shot pulse when the referenced bit changes from 0->1
- Negative Transition Contact (Fall)
  - A one shot pulse when the referenced bit changes from 1->0

Basic Ladder Elements - Coils

- Direct Coil -- turns ON when the preceding net conditions are ON
- Inverted Coil -- turns OFF when the preceding net conditions are ON
- Set Coil -- turns a set coil ON (latches), when preceding net conditions are ON
- Reset Coil -- turns a set coil OFF (unlatches), when preceding net conditions are ON
- Toggle Coil -- changes its state when it is activated
Simple Ladder Logic Program

HMI Editor

- Insert shapes, buttons and images onto the display and associate them with specific elements (operands) from the ladder logic program