L12 - Studio 5000® and Studio 5000 Logix Designer®: Basics Lab

Introducing the Studio 5000 Logix Designer® Environment
Introduction
Studio 5000 Logix Designer®

V20 and back

RSLogix 5000®

V21….and forward

Studio 5000 Logix Designer®
Studio 5000® Components

Intuitive, integrated design and configuration suite with multiple Plugin components
Controller Portfolio

Micro Control Platform
Micro800™ Controller
• Low acquisition cost
• Easy connectivity
• Simple programming tools
• Ideal for standalone machines

Standard Machines
CompactLogix™ Controller
• Multiple control disciplines
• Flexible and scalable
• Real time information-enabled
• Standard, unmodified Ethernet
• One common integrated design environment
• Local and distributed I/O options

Complex Machines & Process
ControlLogix® Controller
• Scalable redundancy for fault tolerance
• Provides safety and availability requirements
• Distributed processing power

Process Safety
AADvance®/Trusted®
• Standard, unmodified Ethernet
• One common integrated design environment
• Local and distributed I/O options
CompactLogix™ Controllers

**CompactLogix™ 5370**
- Integrated Motion on EtherNet/IP up to 16 axes
- Linear and Device Level Ring network topologies for up to 48 nodes
- Integrated safety up to SIL 3, PLe CAT 4 versions
- On-Machine™ version

**CompactLogix™ 5380**
- Integrated Motion on EtherNet/IP up to 32 axes
- Two Ethernet ports for dual IP or support for linear and Device Level Ring topologies for up to 80 nodes
- Enables high-speed I/O, motion control
- Enhanced security features

**CompactLogix™ 5480**
- Enables high-speed I/O, and Integrated Motion on EtherNet/IP up to 150 axes
- Includes three GbE EtherNet/IP ports supporting both linear or Device Level Ring topologies up to 250 nodes
- Provides a Logix based real time controller that runs in parallel to an instance of Windows 10 IoT Enterprise
- Enhanced security features
ControlLogix® Controllers

**ControlLogix® 5570**
- Integrated Motion on EtherNet/IP up to 100 axes
- Integrated safety up to SIL 3, PLe CAT 4 versions
- On-Machine™ versions
- Conformal coat and extreme environment versions
- Redundancy and removal insertion under power

**ControlLogix® 5580**
- Integrated Motion on EtherNet/IP up to 256 axes
- 1 gigabit (Gb) embedded Ethernet port enables high-speed I/O and motion control for up to 300 nodes
- Conformal coat versions
- Removal insertion under power
- Enhanced security features
Integrated Motion, Safety and Drive Control

Controller owns everything

Standard Ethernet via EtherNet/IP

Stratix® config and diagnostics

Safety I/O

Motion & Safety control
Lab Format

- There will be 10 sections. For each section there will be a
  - Presentation
  - Lab
  - Discussion
- Each lab is more or less independent
- Questions?
<table>
<thead>
<tr>
<th>Agenda</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New Project</strong></td>
<td></td>
</tr>
<tr>
<td>Tags</td>
<td>Download</td>
</tr>
<tr>
<td>I/O</td>
<td>UDTs</td>
</tr>
<tr>
<td>Programming</td>
<td>AOIs</td>
</tr>
<tr>
<td>Tasks and Programs</td>
<td>Trends</td>
</tr>
</tbody>
</table>
New Project

Controller Selection

- Project Window
- Controller organizer
- Instruction Toolbar
- Controller Faceplate

Logix Designer

CompactLogix™ & ControlLogix®

Memory sizes

“ACD” file
New Project

Controller Selection

- Project Window
- Controller organizer
- Instruction Toolbar
- Controller Faceplate

Revision
Chassis Size
Slot

New Project

1756-L71 ControlLogix® 5570 Controller

Revision: 31
Chassis: 1756-A10 10-Slot ControlLogix Chassis
Slot: 0
Security Authority: No Protection
Secure With: Logical Name <Controller Name>
Description:
Redundancy: Enable

Cancel  Back  Next  Finish
New Project

Controller Selection

Project Window

Controller organizer

Instruction Toolbar

Controller Faceplate

Instruction Palette 250+ predefined instructions

Controller Organizer

Programming Area

Quick View
New Project

Controller Selection
Project Window
Controller organizer
Instruction Toolbar
Controller Faceplate

Descriptive Tags
Integrated safety
Integrated motion
Integrated Alarming
Definitions
I/O

Controller Organizer
- Controller MyGuard
  - Controller Tags
    - Controller Fault Handler
    - Power-Up Handler
  - Tasks
    - MainTask
      - MainProgram
    - SafetyTask
      - SafetyProgram
      - Unscheduled
  - Motion Groups
  - Alarm Manager
  - Assets
    - Add-On Instructions
    - Data Types
    - Trends
  - Logical Model
    - I/O Configuration
      - 1756 Backplane, 1756-A10
        - [0] 1756-L84ES MyGuard
      - Ethernet
        - 1756-L84ES MyGuard
New Project

- Controller Selection
- Project Window
- Controller organizer
- Instruction Toolbar
- Controller Faceplate
- Controller Status
- Online/Offline
- Online Program Edits
- Controller Properties
New Project

- Lab 1 – Creating a project
  - 2 minutes
Discussion

- What are some controller types supported by the software?
- What is the main area in a project?
New Project

Discussion

- What are some controller types supported by the software?
  - ControlLogix®, GuardLogix®, CompactLogix™, Compact GuardLogix®
- What is the main area in a project?
  - Controller Organizer
- Questions?
Agenda

New Project
Download

I/O
UDTs

Tags
AOIs

Programming
Logical Organizer

Tasks and Programs
Trends
A controller can handle many (different) devices
Some I/O families
Remote I/O
Controller Organizer
Ethernet Adapter (Node)
Local I/O

I/O Configuration folder
Adding I/O Profiles
I/O Profile Properties

Modules and Devices
Each remote rack or device is a node (5580 and 5380)

Objects in the I/O tree are called “Profiles”
I/O

Modules and Devices

I/O Configuration folder

Adding I/O Profiles

I/O Profile Properties

Search by part number

Manufacturer filter

I/O Type filter

List of available modules

Do we want to add more modules?
Three Choices for Keying

- **Exact Match**
  - Catalog and Revision must exactly match
  - Regulated Industries

- **Compatible Module**
  - Module will determine if it can support features
  - This is recommended setting

- **Disable Keying**
  - Always try to connect regardless of match.
  - Only use if manufacturer advises.
I/O Configuration folder

Adding I/O

I/O Profile Properties

Module Info

Multiple tabs

Information/config area

Configuration

Other Options

[Diagram showing a software interface for I/O configuration with multiple tabs and configuration options.]
Lab 2 – Adding I/O
  - 2 minutes
Discussion

- Where is I/O located in a project?
- What is keying?
I/O

Discussion

- Where is I/O located in a project?
  - I/O Configuration folder
- What is keying?
  - Determines if a module is compatible with profile
- Questions?
Tags

- I/O Tags
  - Only I/O Tags have colons in the tag name
  - Local modules begin with "Local"
  - Remote modules begin with the adapter name
- Controller Scope
- Controller vs Program Scope
- Adding Tags
- Alias Tags

Controller Tags - IOTest (controller 1)

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Force Mask</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node6 IO1:1:Data2</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node6 IO1:1:Data3</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node6 IO1:1:Data4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node6 IO1:1:Data5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node6 IO1:1:Data6</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node6 IO1:1:Data7</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node6 IO1:1:C</td>
<td>(...)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local2:1</td>
<td>(...)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local2:C</td>
<td>(...)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local1:1</td>
<td>(...)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local1:C</td>
<td>(...)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Controller Scope manages tags accessible within the controller's memory, while other scopes manage tags that are specific to the program.
Tags

• Data Types
  • Determines what kind of data the tag hold
  • Atomic types
    • BOOL, SINT, INT, DINT, REAL, STRING
  • Structures
    • Examples
      Timer, Counter, Message, UDT
Tags

- Arrays

- Viewing Tags
- I/O Tags
- Adding Tags
- Controller vs Program Scope
- Alias Tags
Tags

Viewing Tags

I/O Tags

Adding Tags

Controller vs Program Scope

Alias Tags

New Tag (Blank Line)

Select Edit Tags Tab

Tag Data Type

Scroll to Bottom
Tags

- **Controller scoped tags**
  - Accessible by any program (Global in scope)
- **I/O module tags**
- **Produce & Consume tags**

Any program in the controller can access these tags.
• Program scoped tags
  • Local in nature
  • Local tag names can be reused
  • Makes code more modular

Only the routines of the same program can access local tags
Tags

- Viewing Tags
- I/O Tags
- Adding Tags
- Controller vs Program Scope

Alias Tags

- **Alias tag**
- **Aliased to?**
- **Alias of an Alias?**
Lab 3 - Tags

2 minutes
Discussion

- Where are I/O tags located?
- What are alias tags?
Discussion

- Where are I/O tags located?
  - Controller scope

- What are alias tags?
  - Alias tags are an alternate name for another tag to allow a more relevant name, such as for an I/O point or an array member

- Questions?
Programming

Routines
- Ladder
- Structured Text
- Function Blocks
- SFC

Code is stored in routines

Task Folder
Task
Program
Program tags
Programming

- Routines
- Ladder
- Structured Text
- Function Blocks
- SFC

- Input Instructions on left of rung
- Output instructions on right of rung

Rungs (of a ladder)
Programming

- **Routines**
- **Ladder**
- **Structured Text**
- **Function Blocks**
- **SFC**

Similar to Pascal (C or Java like)

Typically one instruction per line

Conditional sections
Programming

- **Steps**
- **Transitions**
- **Actions (in STX)**

**SFC**

**Structured Text**

**Function Blocks**

**Ladder**

**Routines**
• Note: There are different flavors of Logix Designer
• Not all versions offer STX, FBD, SFC
• Professional Edition has all languages
  • This is what we will be using
Programming

- Lab 4 – Adding Logic
  - 2 minutes
Programming

Discussion

- What are some of the available languages?
- What object holds code?
Discussion

- What are some of the available languages?
  - Ladder, FBD, STX, SFC
- What object holds code?
  - Routines
- Questions?
Tasks and Programs

- Three types of tasks
- Tasks control when code runs
  - Should use tasks to run code only when necessary
Tasks and Programs

- Continuous task
- Runs in free time
Tasks and Programs

- Periodic task
- Runs per defined period
Tasks and Programs

- Event task
- Runs per defined Event

Task Properties - Demo_Event_Task

- Type: Event
- Trigger: Module Input Data State Change
- Tag: Local:3:1
- Priority: 10
- Watchdog: 500.000 ms

Configurable Event
Optional Execution
Priority
Tasks and Programs

Tasks hold programs. All programs execute when task does.

Programs hold routines.

Main routine is defined under program properties and has a “1” for 1st.
Tasks and Programs

- Programs should be used to modularize code parts of a:
  - Machine
  - Process
  - System

Tasks

- Continuous
- Periodic
- Event

Programs

Multiple programs
Tasks and Programs

- **Routines** can then be used to further modularize code for sub parts of a machine, etc.
Tasks and Programs

- Lab 5
  - 2 minutes
Tasks and Programs

Discussion

- What types of tasks are available?
- What are programs?
Tasks and Programs

Discussion

- What types of tasks are available?
  - Continuous, Periodic, Event

- What are programs?
  - Programs hold routines, have a defined main routine, and are assigned to tasks

- Questions?
<table>
<thead>
<tr>
<th>Agenda</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Project</td>
<td></td>
</tr>
<tr>
<td>I/O</td>
<td>UDTs</td>
</tr>
<tr>
<td>Tags</td>
<td>AOIs</td>
</tr>
<tr>
<td>Programming</td>
<td>Logical Organizer</td>
</tr>
<tr>
<td>Tasks and Programs</td>
<td>Trends</td>
</tr>
</tbody>
</table>
Download

- Terms - Download

A download compiles the ACD file and loads it into the controller.
### Terms - Upload

An upload reads the controller memory and reconstructs the ACD file.
We will be using the EtherNet/IP driver.

RSLinx® has a selection of available drivers.
Who Active

- uses RSLinx®
Key switch Position
- Program
- Remote
- Run
- Remote Position
  - Run/Program controlled by software

Use drop down to change mode
Download

Who Active

Mode

Run

Faceplate and controller LED show run

Green power rails

Buttons and instructions reflect live state
Download

- Online edits
  - Except if keyswitch is in “Run”
Download

- Lab 6 – Download and test the program
  - 5 minutes

- Open this ACD file

  C:\Lab Files\Basics Logix\Logix_Basics_Lab_Demo_Project.ACD
Discussion

- What is online? What is offline?

- What is indication controller is running?
Discussion

- What is online? What is offline?
  - Offline is not monitoring controller, online shows live data and states
- What is indication controller is running?
  - Controller LED. When online - faceplate, green rails
- Questions?
User Defined Type

- How can I keep data about an object together in one place?
- What if I have 1000 cars? (That a lot of tags!)

- Color is Red
- Basic model
- Weight is 2 Kilos
- 17 inch rims
- 2 doors
User Defined Type

- **User Defined Type**
  - Made of simpler types
  - **Organizes** related information
  - Defined under Data Types folder
User Defined Type

- Create a UDT type with desired members
  - Good practice - Use descriptions
User Defined Type

- Once a UDT is defined, a tag can be created with this type
User Defined Type

- Tag can be expanded to see members
User Defined Type

- A UDT can be used as an array
- A UDT can contain arrays or other UDTs
User Defined Type

<table>
<thead>
<tr>
<th>UDT</th>
<th>Define</th>
<th>Tag</th>
<th>Array</th>
</tr>
</thead>
</table>

- Tags appear in instructions just like they do in the tag list

Car1_Data.Weight is the same in tags as in instructions
User Defined Type

- Lab 7 – UDT

2 minutes
UDT Discussion

- Why use a UDT?
- Can a UDT contain other UDTs?
UDT Discussion

- Why use a UDT?
  - For simple and easy organization and documentation of related data
- Can a UDT contain other UDTs?
  - Yes!
- Questions?
Add-On Instruction

Concepts and Guidelines

Parameters and Tags

Code

Instantiation

Samples
Key Idea: Always use a systematic approach to solving problems

State the Problem
Define Input & Outputs
Develop Algorithm
Solve Problem
Test Solution

Proper planning can save time and money in the long run!
Add-On Instruction

- Modular program design
  - Well defined interface
  - Well defined function
- Defined once
  - Used many
- Should have a clearly defined purpose
Add-On Instruction

Concepts and Guidelines

- **Not** editable online
  - Needs to be **testable** and manageable with a reasonable amount of code

- Single Routine
Add-On Instruction

- When creating or changing an AOI
  - Give it a name, rev, and who created it.
Add-On Instruction

Concepts and Guidelines

Parameters and Tags

Code

Instantiation

Samples

Parameter List

Input
(Uses copy of data)

Output
(Updates data when done)

InOut
(Direct tag memory reference)
Add-On Instruction

- Local Tags
  - Local to the AOI
  - Not accessible by outside ladder

Local tag list
Add-On Instruction

Concepts and Guidelines

Parameters and Tags

Code

Instantiation

Samples

Written in any language

Used in any language

// Oil feeder valve
Value(Value01, InputPoint, OutputPoint, Value01Config);
Add-On Instruction

• Each AOI needs a **unique backing tag**
Add-On Instruction

Concepts and Guidelines

Parameters and Tags

Code

Instantiation

Samples
Add-On Instruction

- PlantPax objects
- Source of AOIs and Faceplates for:
  - Device control (like valves)
  - Hart modules
  - Process instructions
AOI

- Lab 8 – AOI

2 minutes
AOI Discussion

• Why use an AOI?

• What are the parts of an AOI?

• Questions? Comments?
AOI Discussion

• Why use an AOI?
  • Encapsulate common code and well defined function that can easily be reused with a single instruction

• What are the parts of an AOI?
  • Input, output, and local tags, with one routine

• Questions? Comments?
<table>
<thead>
<tr>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tasks and Programs</strong></td>
</tr>
<tr>
<td>New Project</td>
</tr>
<tr>
<td>Programming</td>
</tr>
<tr>
<td>Tags</td>
</tr>
<tr>
<td>I/O</td>
</tr>
<tr>
<td>Tasks and Programs</td>
</tr>
<tr>
<td><strong>Download</strong></td>
</tr>
<tr>
<td>UDTs</td>
</tr>
<tr>
<td>AOIs</td>
</tr>
<tr>
<td><strong>Logical Organizer</strong></td>
</tr>
<tr>
<td>Trends</td>
</tr>
</tbody>
</table>
A Logical View of the System

VFFS
Infeed
Forming
Filling
Sealing
Mixer
Addition
Agitate
Pressure Ctrl

VFFS
Infeed
Forming
Filling
Sealing
Mixer
Addition
Agitate
Pressure Ctrl

Filling
Forming
Sealing
Why Add A Logical Organizer?

- To create an organizational model of the system from the user’s perspective.
- Helps facilitate good modular automation design practices.
- Paradigm shift from how the “controller executes code” to how the “user views the system”.

Programs Span the Tasks Logically Organized
System Organizer Logical View
Library .ACD Deployment

Create from Library

Add to Library

This includes Dependencies (UDT, AOI's)
Library Options

- Using XML Code
  - Import/Export
  - Library Management Tools

Right Click
Logical Organizer

- Lab 9 – Logical Organizer

1 minute
Logical Organizer Discussion

• Why use the Logical Organizer?

• Does rearranging the Logical Organizer affect the Controller Organizer?

• Questions? Comments?
Logical Organizer Discussion

• Why use the Logical Organizer?
  • Group related code and for Library functions.

• Does rearranging the Logical Organizer affect the Controller Organizer?
  • No!

• Questions? Comments?
<table>
<thead>
<tr>
<th>Agenda</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>New Project</td>
<td>Download</td>
</tr>
<tr>
<td>Programming</td>
<td>UDTs</td>
</tr>
<tr>
<td>Tags</td>
<td>AOIs</td>
</tr>
<tr>
<td>I/O</td>
<td>Logical Organizer</td>
</tr>
<tr>
<td>Tasks and Programs</td>
<td><strong>Trends</strong></td>
</tr>
</tbody>
</table>
- Trends graph data
  - Part of project file
  - Atomic Data types
  - Samples as fast as 1ms for fast applications
Trends

- Running Trend
- Adding a trend
- Properties
- Start Stop triggers

**Scroll Buttons**

**Run/Stop**
Trends

- Many trends can be created
  - Unique name
  - Sample period

**Adding a trend**

![New Trend dialog box](image)
Trends

• Display properties

Very configurable
• Display properties

Can have multiple tags
Trends

- Config - Start and Stop Triggers

Extra Samples

Condition
Trends

- Example

![Diagram of Trend Analysis with Start and Stop Triggers]

- Trends
- Running Trend
- Adding a Trend
- Properties

Start Stop triggers
Lab 10 – Trends

4 minutes
Trends Discussion

- How fast can a trend sample?
- Can there be more than one tag?
Trends Discussion

- How fast can a trend sample?
  - 1 Millisecond
- Can there be more than one tag?
  - Yes!
- Questions?
Modular Features

• Modular Program Design
  • Add-On Instructions (AOIs)
  • User-Defined Types (UDTs)
  • Program Parameters
Modular Features

- Code and Machine Management
  - Import/Export
  - Drag and Drop Environment
  - Library management
  - Logical Organizer
Questions ?
Thank You!
Lab 1
  2 minutes
Timer Template

- Lab 1
  - 5 minutes