



ENGINEER
NEXT

NIDays

The image features a background of diagonal stripes in various shades of blue, green, orange, and red. The text 'ENGINEER NEXT' is prominently displayed in white, with 'ENGINEER' in a smaller font above 'NEXT'. A yellow graphic element, resembling a stylized 'X' or a folded ribbon, is positioned between the two words. Below this, the text 'NIDays' is enclosed in a white rectangular box with a thin border.



Corrado Degl'Incerti Tocci

Staff Applications Engineer

Certified LabVIEW Architect (CLA)
NI Certified Professional Instructor

Today's Objective :

Help you focus your preparation
to increase your confidence and your
readiness
for successful CLD certification

Presentation Agenda: CLD Exam

- Preparation
- Objective and theme
- Design your way to success
- Process, format and scoring
- Tips, Conclusion of exam

Preparation Resources



Download the Preparation E-kit for the NI Certified LabVIEW Developer (CLD) Exam

Thank you for your interest in NI training and certification. The following resources can help you prepare for the Certified LabVIEW Developer (CLD) exam.

Exam Details

- Exam format: Application development
- Prerequisite: Certified LabVIEW Associate Developer
- Exam duration: Four hours

Preparation Resources for the CLD Exam

- [CLD Preparation Guide \(PDF\)](#)
- [Preparation: Download Certification Prep Exam Sample Exercises](#)
- [Certified LabVIEW Developer Preparation Videos](#)
- [CLD Sample Exams](#)

www.ni.com/cld-prep



NI CONFIDENTIAL

CLD Exam Preparation: Effective Exam Preparation



Take Sample Exams
Check on the forums
Read NI.COM papers
Review course materials
Some exercises?



Certification Forum

Preparation Guide

Success Package

Sample Exams

Steps



Certification Forum

Preparation Guide

Success Package

Sample Exams

Steps

CLD Exam Objective & Theme

1. CLD Exam Objective
2. CLD Exam Theme


CLD Exam Objective:


CLD is the Middle of 3 LabVIEW certifications

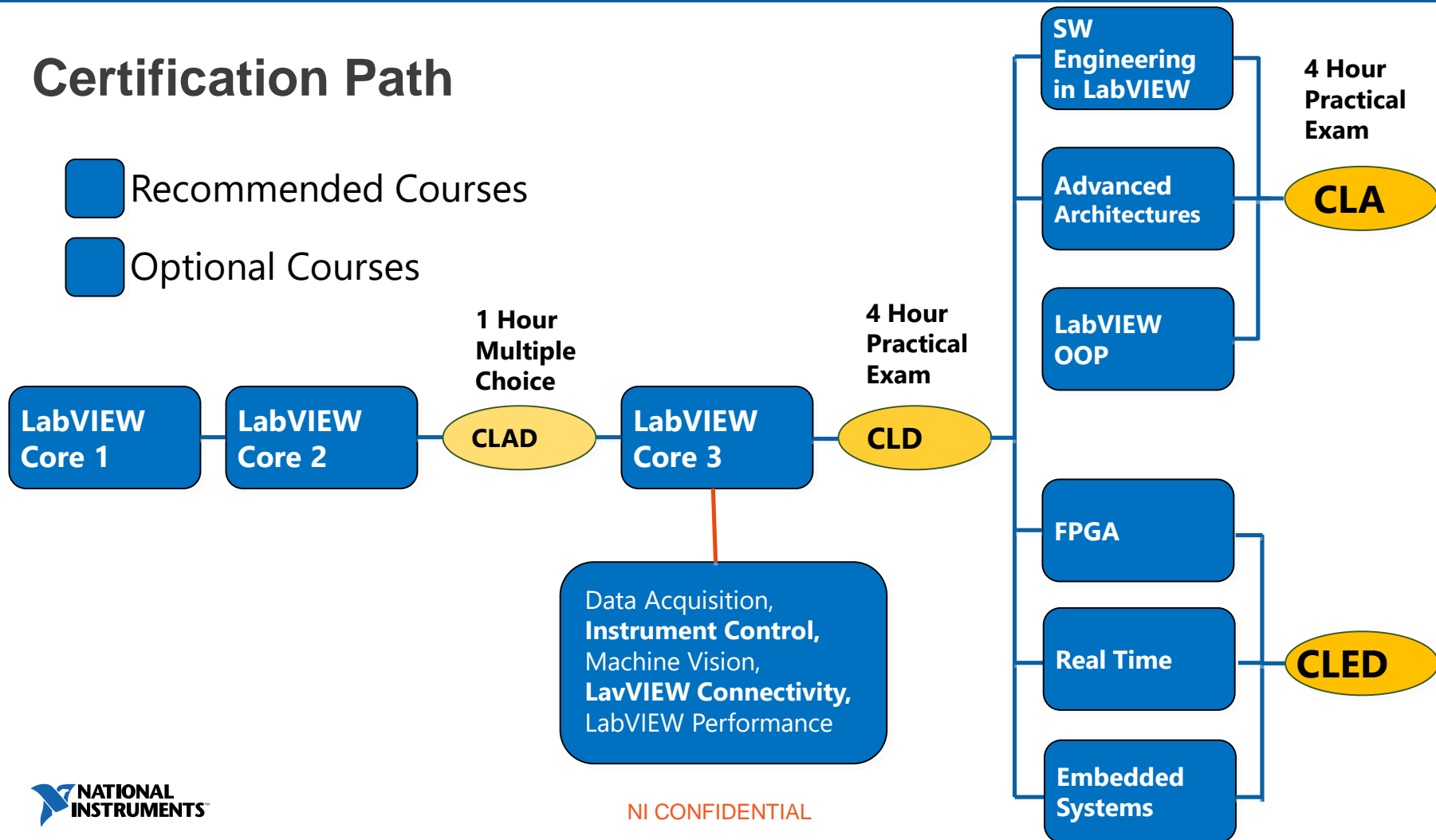
- Certified LabVIEW Associated Developer (CLAD)
 - Working knowledge of LabVIEW
- Certified LabVIEW Developer (CLD)
 - Able to plan & produce stand-alone applications
- Certified LabVIEW Architect (CLA)
 - Able to lead a team developing large applications or a suite of interacting applications.

Certification overview details at <http://sine.ni.com/nips/cds/view/p/lang/en/nid/201888>

Certification Path

 Recommended Courses

 Optional Courses



What skills does a CLD demonstrate?

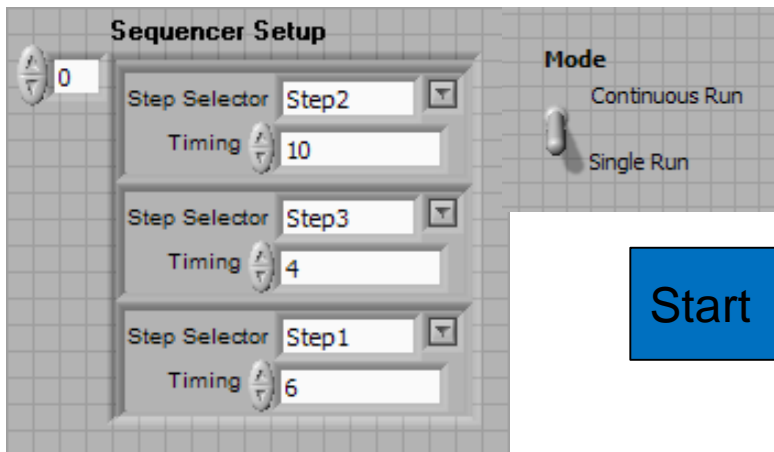
- Problem solving skills
- LabVIEW competency
- Modular, scalable, and maintainable application design experience
- Consistent documentation
- Moderate development (wiring) speed
- Debugging and testing

CLD Exam Objective & Theme

1. CLD Exam Objective

2. CLD Exam Theme

CLD Exam Theme: Sequencer Scenario



Start

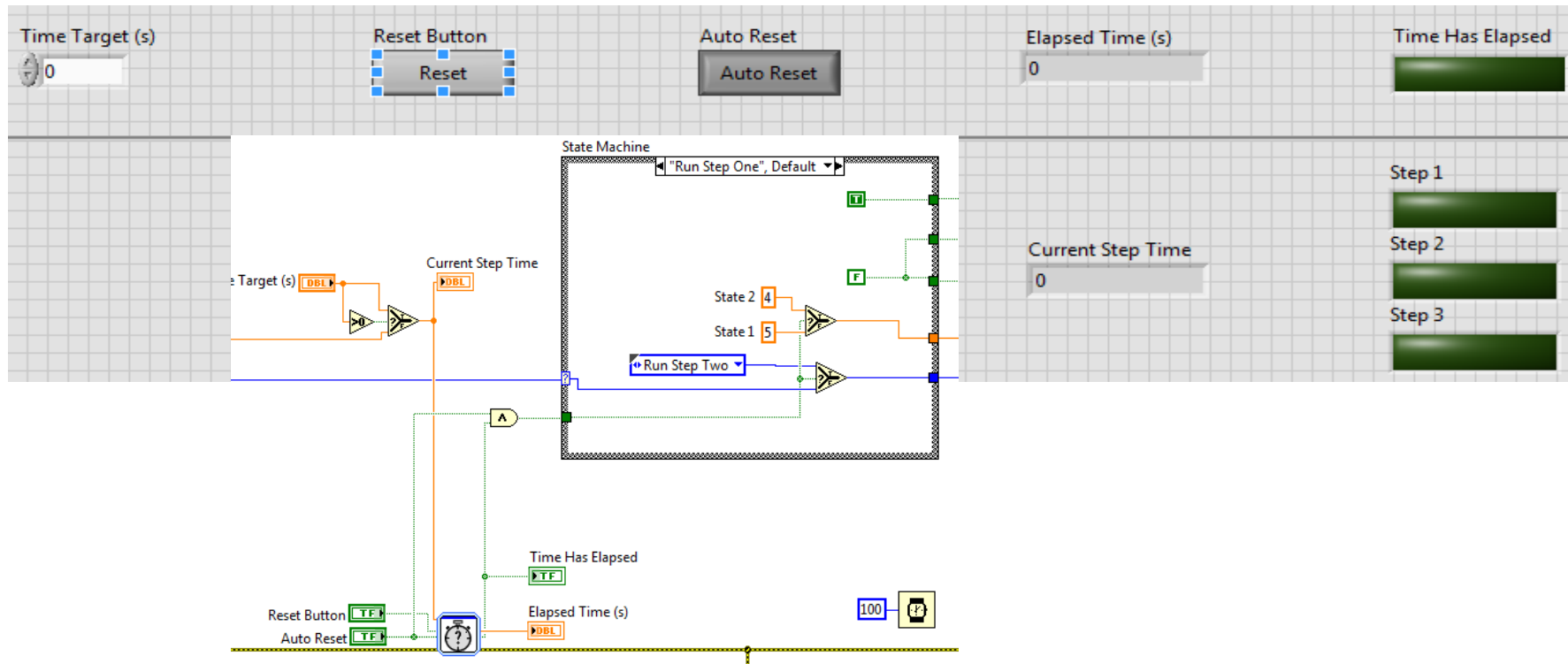
Step 1
Execute: x secs

Step 2
Execute: y secs

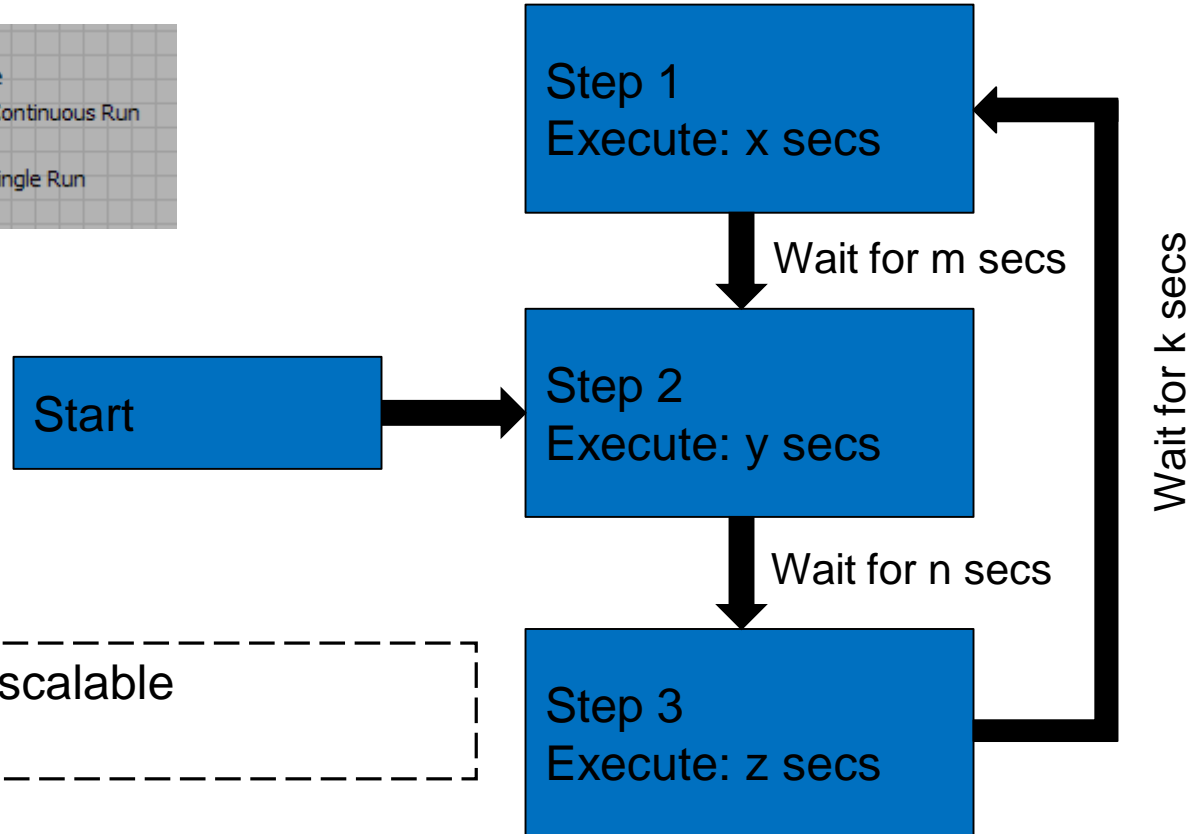
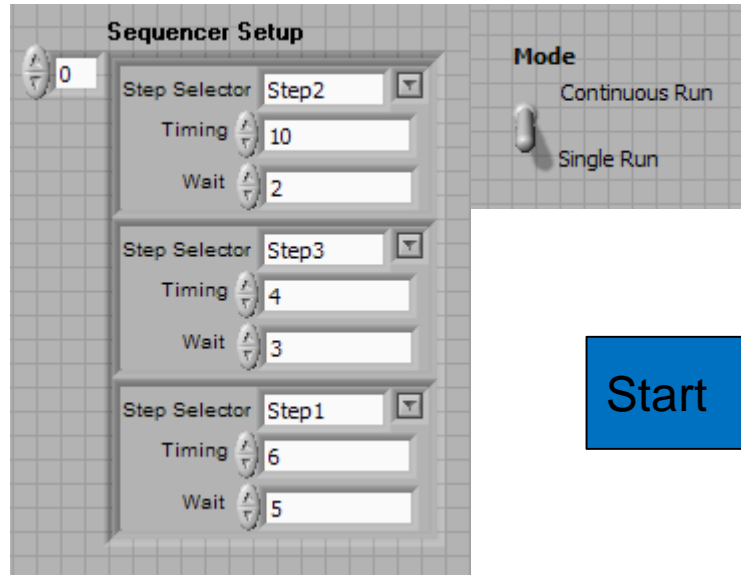
Step 3
Execute: z secs

Program design should be scalable
to adapt to additional steps

Example Sequencer VI from Success Package #9

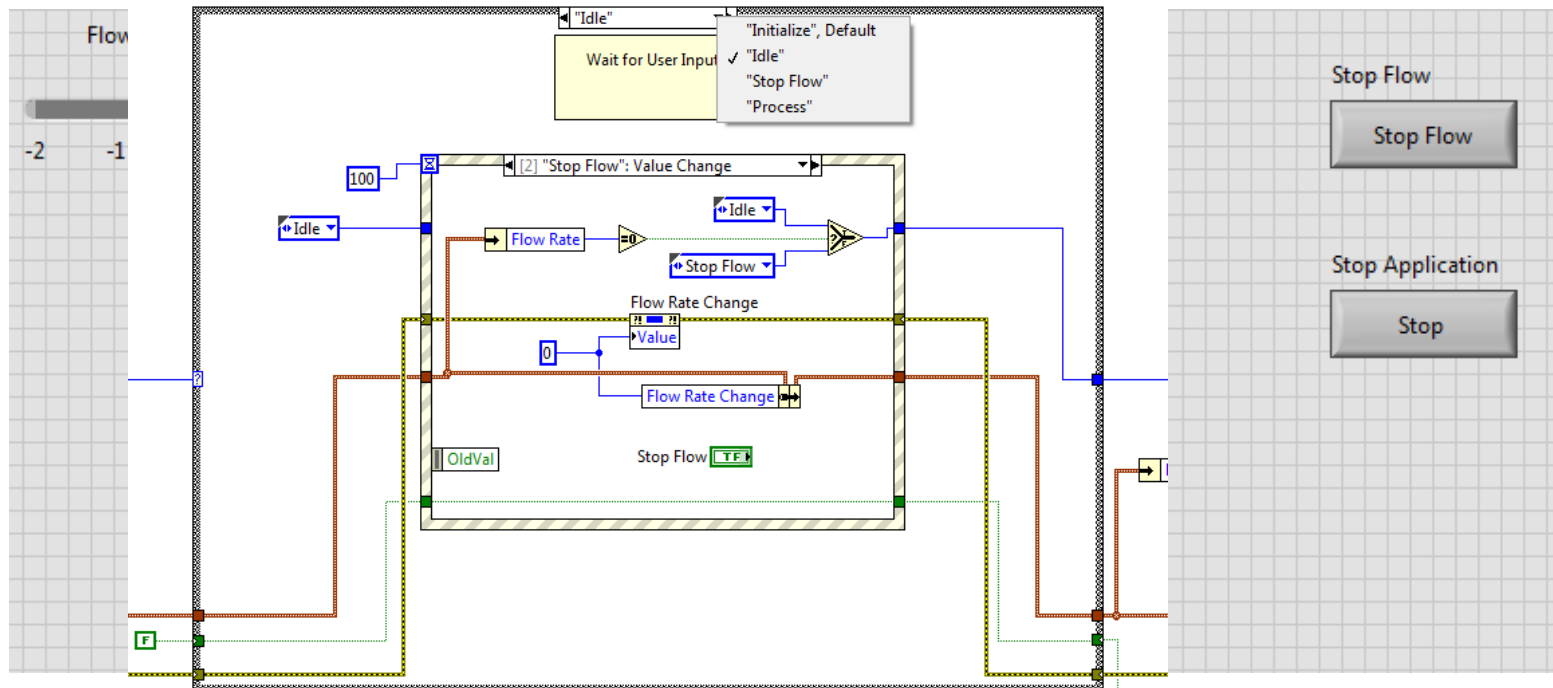


CLD Exam Theme Scenario – Sequencer (with wait)

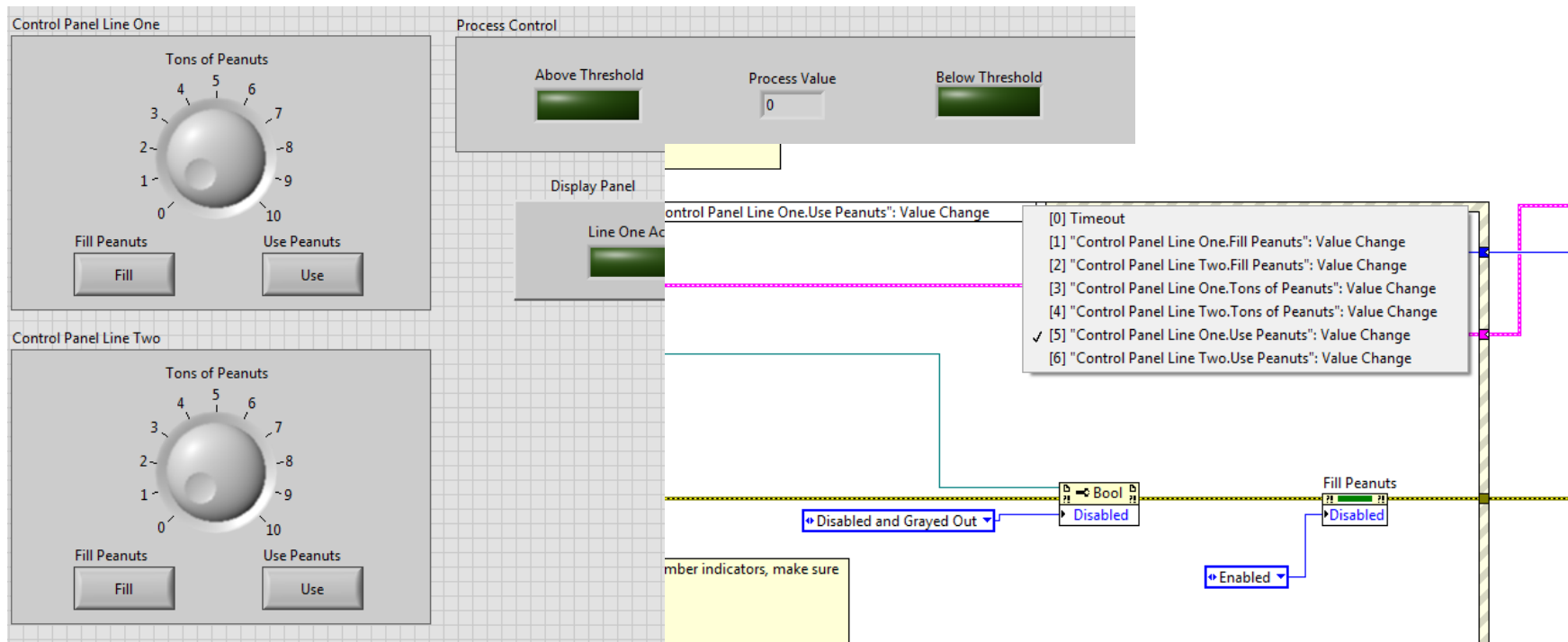


Program design should be scalable
to adapt to additional steps

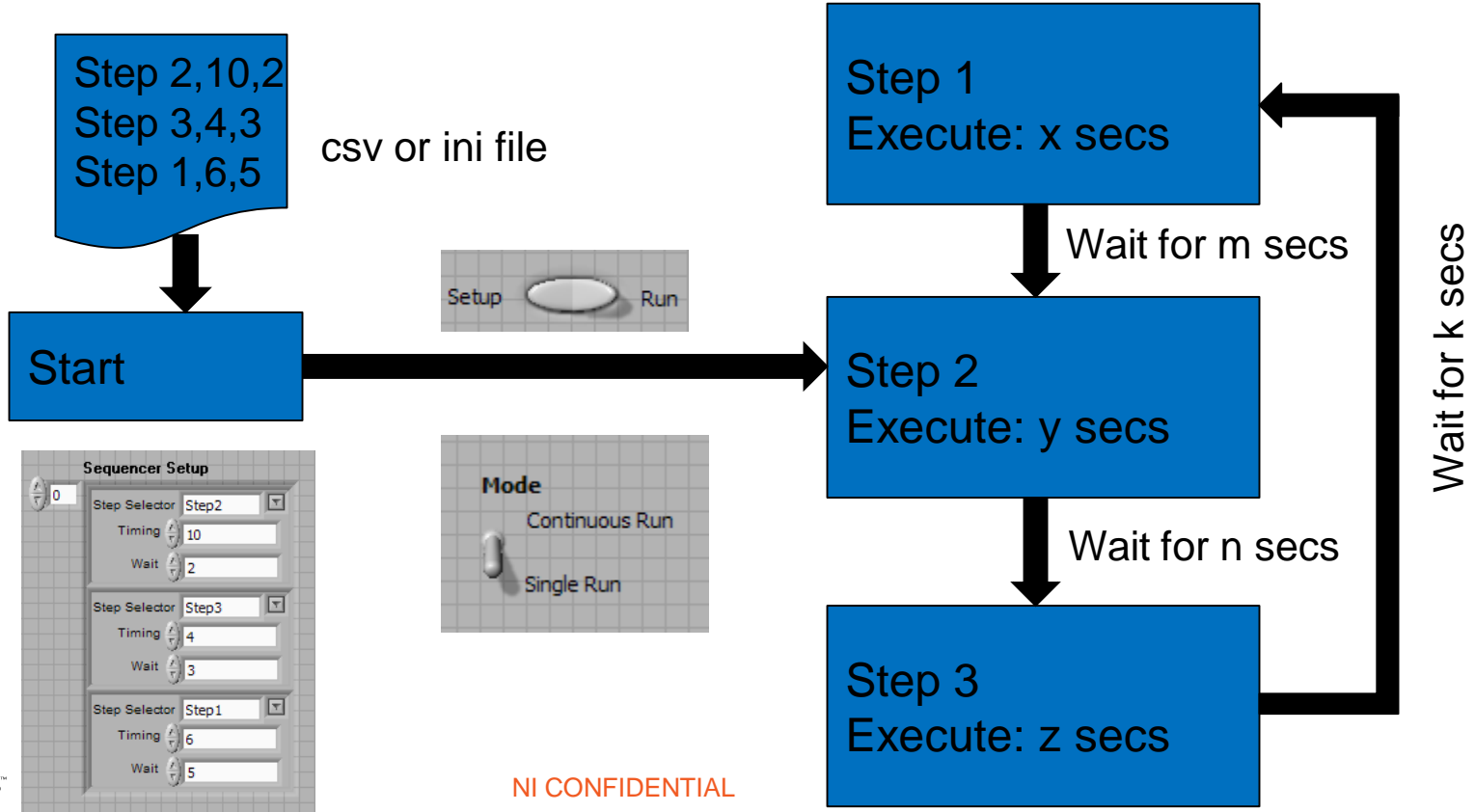
Example Sequencer VI from Success Package #13



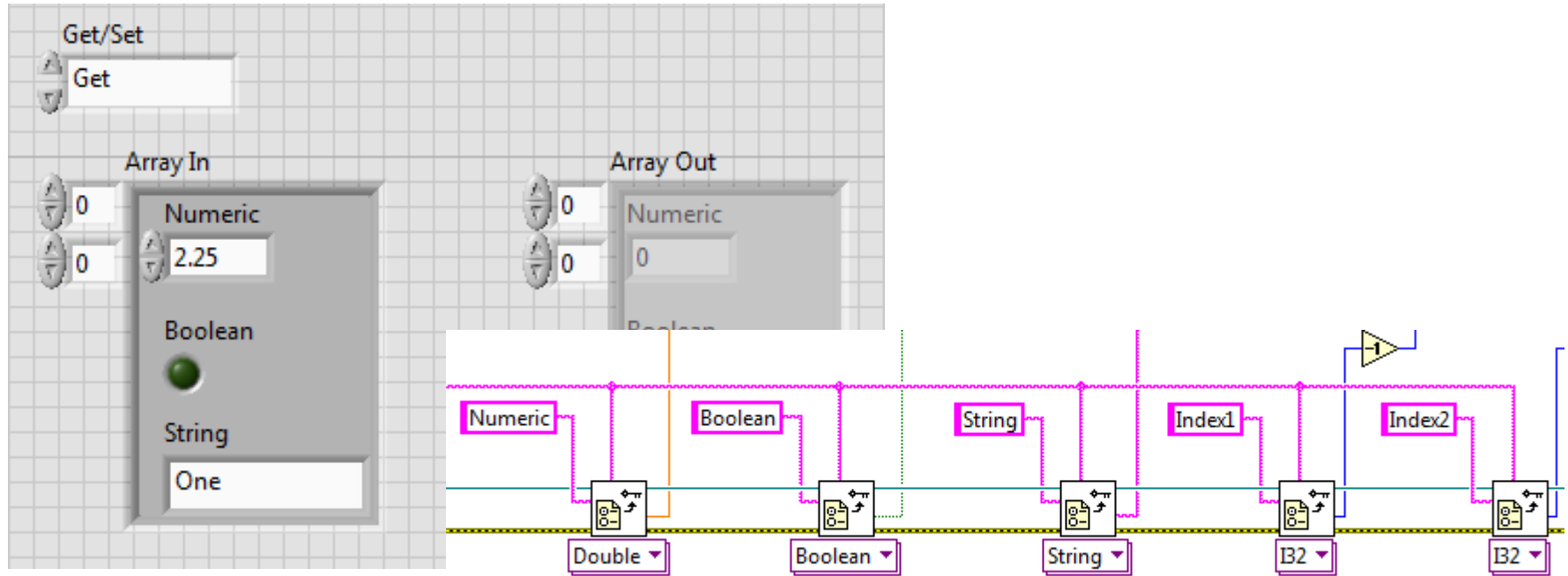
Example Sequencer VI from Success Package #16



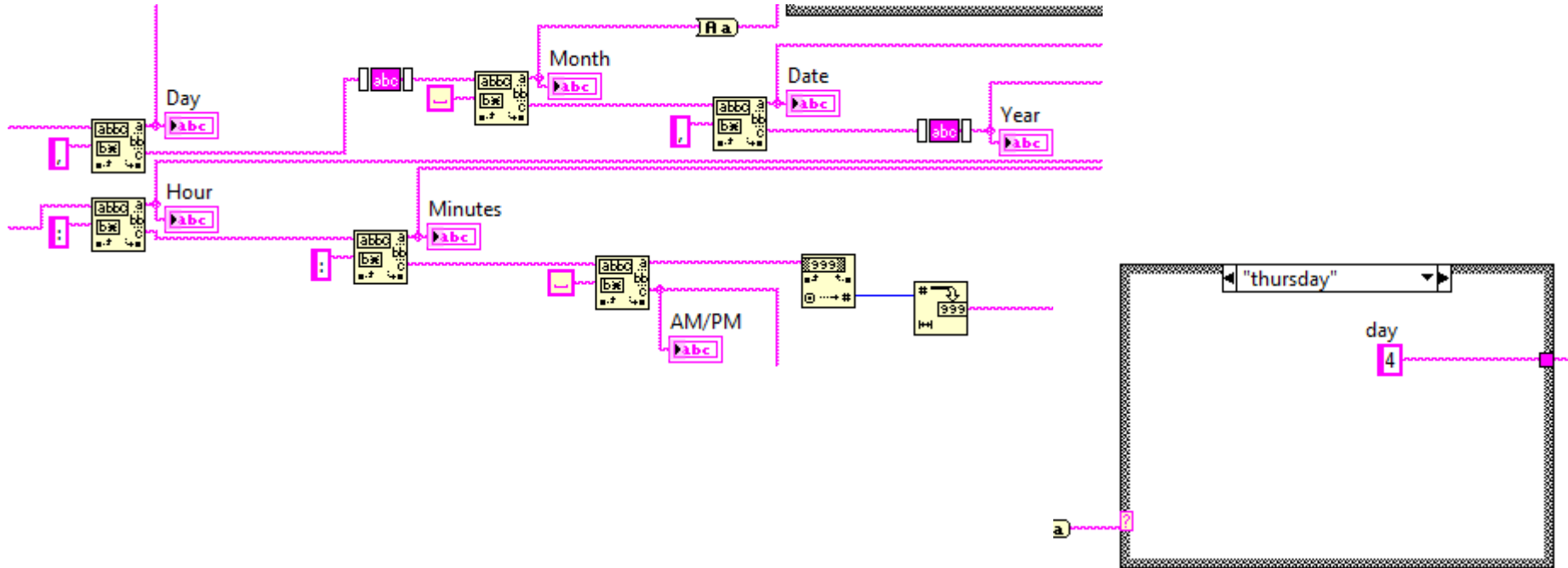
CLD Exam Theme Scenario: Sequencer (with wait & file IO)



Example File IO from Success Package #5

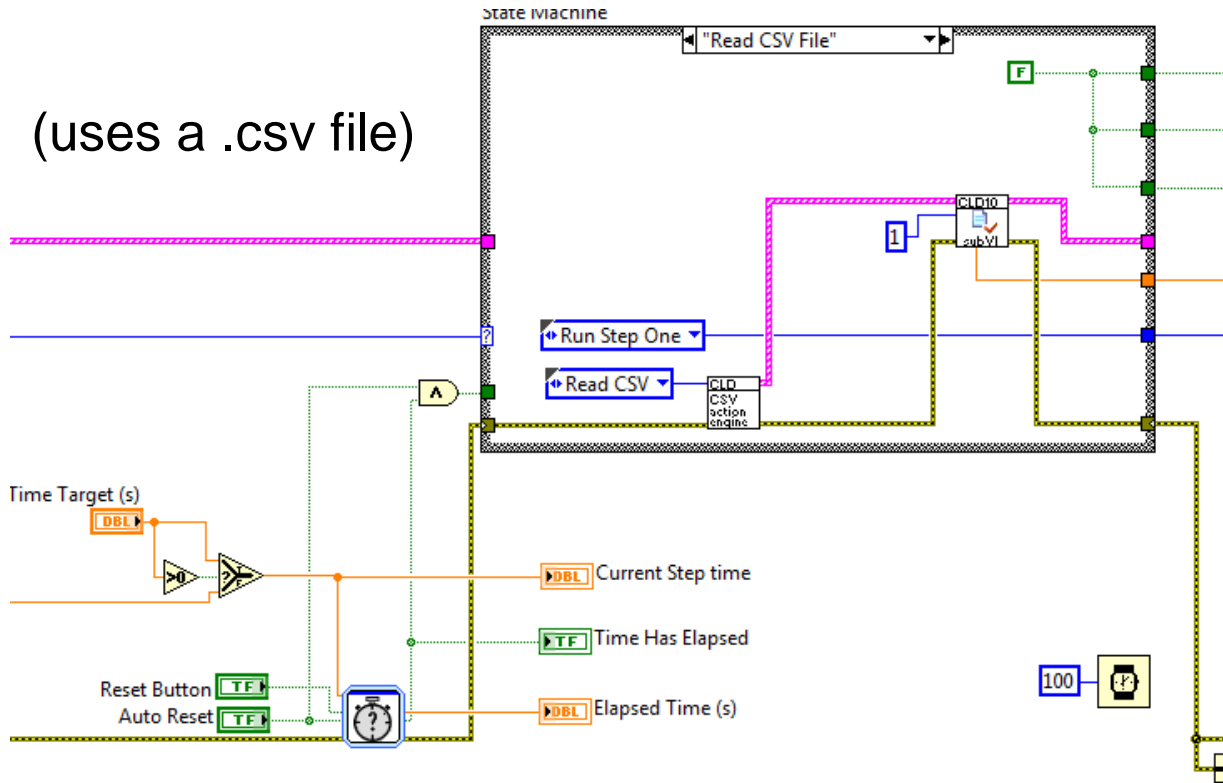


Example String Parsing VI from Success Package #7

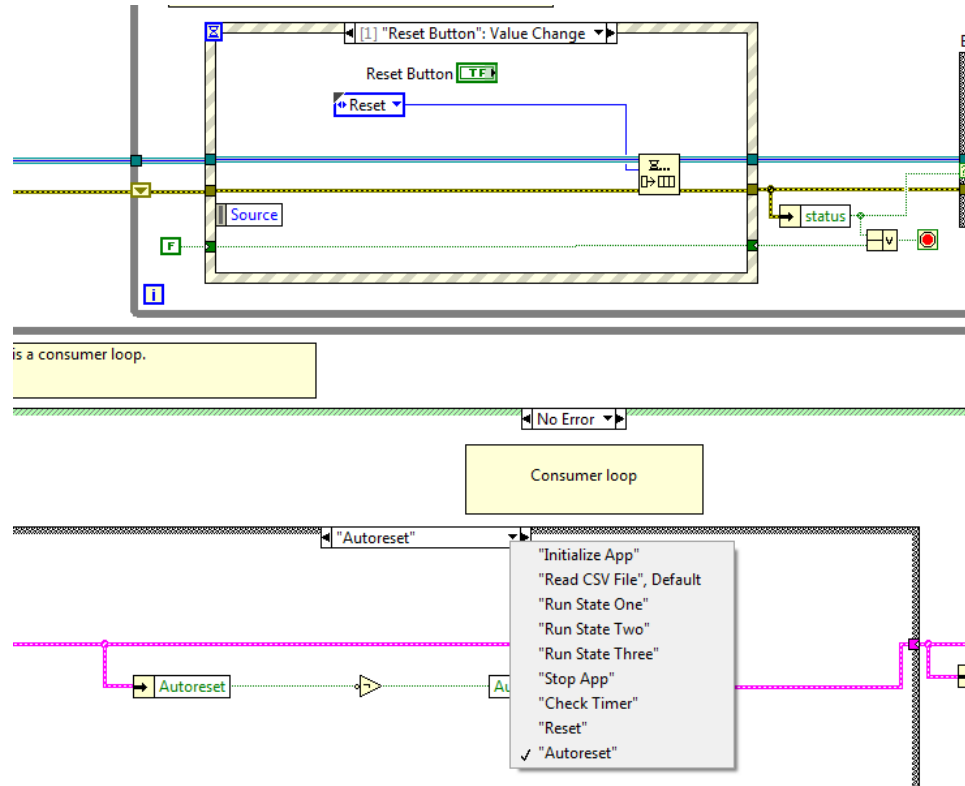


Example Sequencer VI w/File IO from Success Package #10

(uses a .csv file)



Example Sequencer VI w/File IO & UI from Success Package #11



CLD Exam Theme - Summary

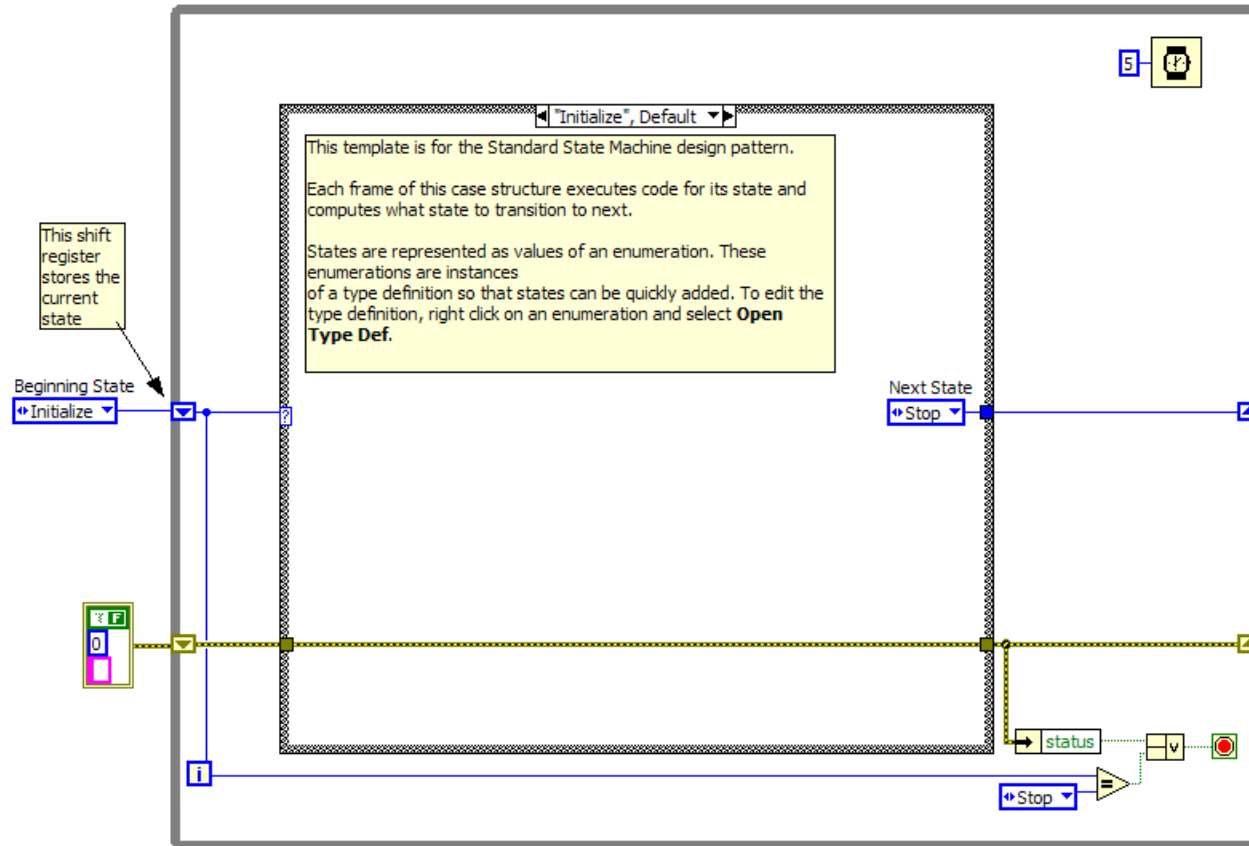
- Exams are based on a sequencer type application
- Step order is configurable
 - via an input file
 - and/or via user interface
- Timing is an essential aspect of the application
- Output or log file may be required
- Application must respond to UI activity within 100 msec.

Design your way to success:

1. Design Patterns

- 2. Timing Methods
- 3. File IO
- 4. Error Handling
- 5. Documentation
- 6. Development Style

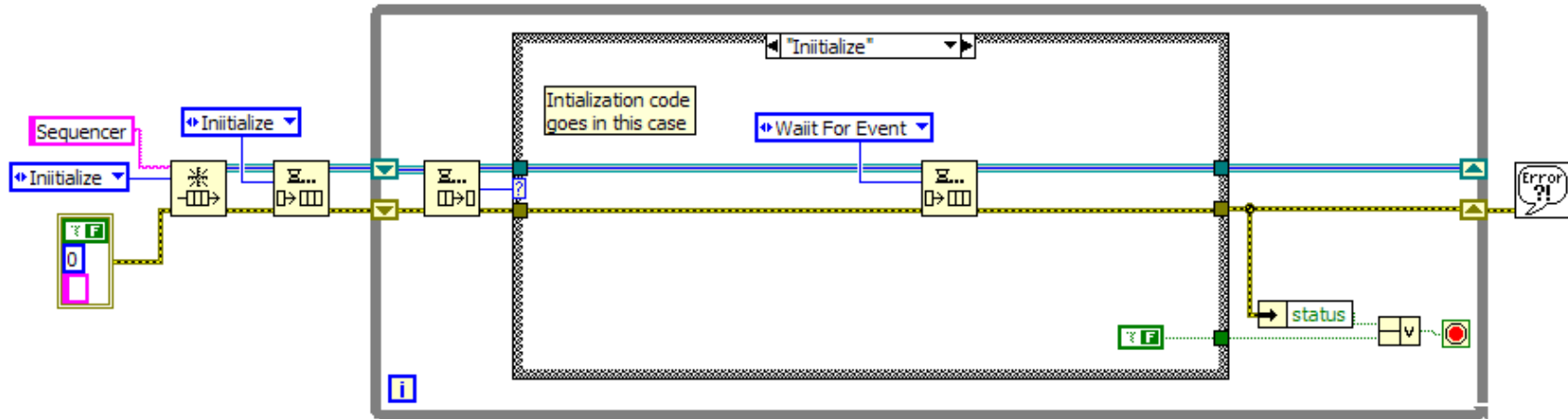
Design Pattern: State Machine



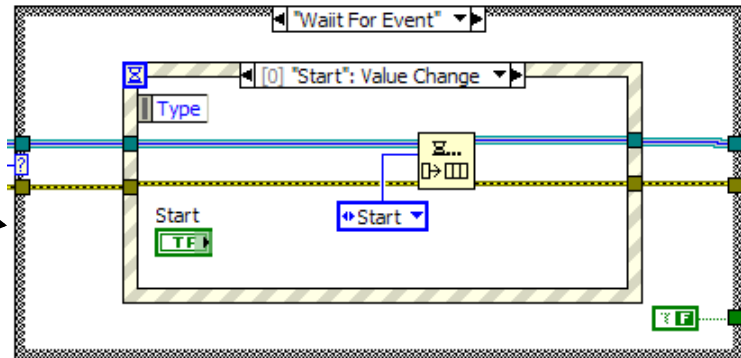
Design Patterns for CLD Applications

Design Pattern	Advantage	Disadvantage
State machine	<ul style="list-style-type: none">• Handles sequence control	<ul style="list-style-type: none">• Cannot handle storing of sequences• May not be responsive enough to user interface events

Design Pattern: Queue Based UI Event Handler



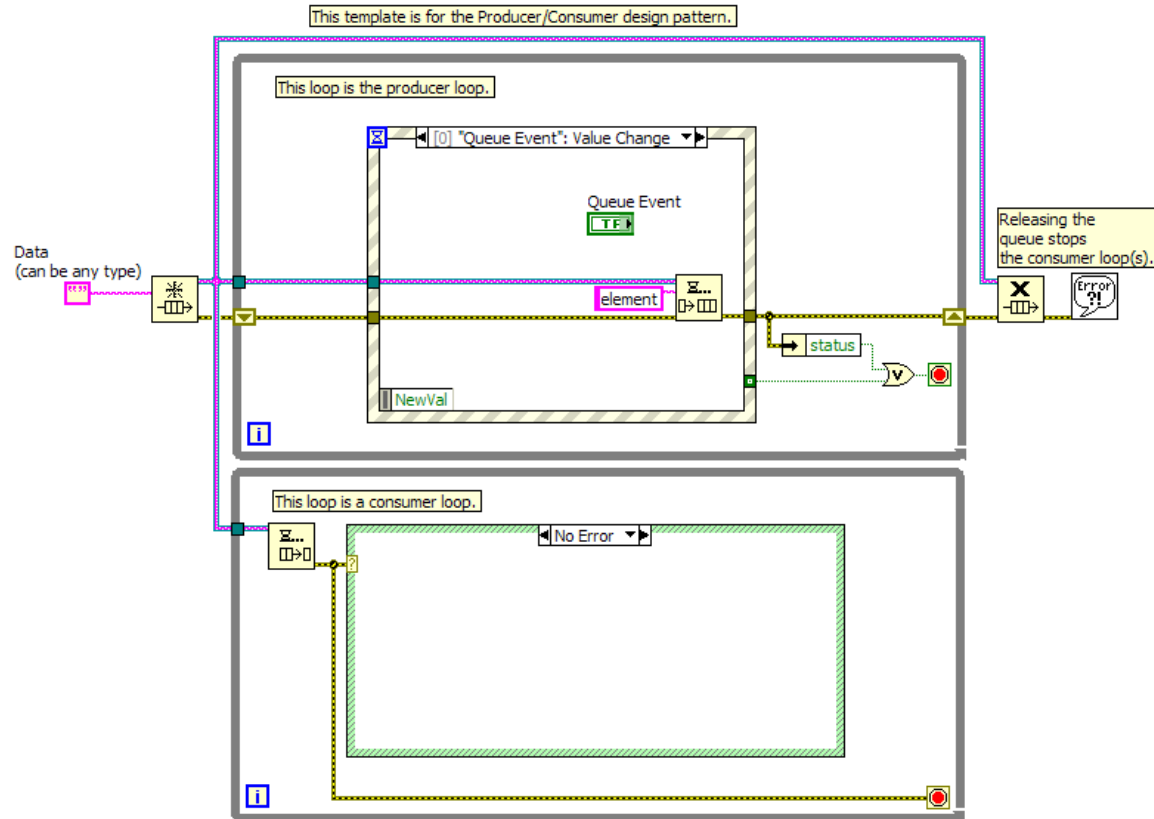
Case with an Event Structure



Design Patterns for CLD Applications

Design Pattern	Advantage	Disadvantage
State machine	<ul style="list-style-type: none">• Handles sequence control	<ul style="list-style-type: none">• Cannot handle storing of sequences• May not be responsive enough to user interface events
Queue based user interface Event Handler	<ul style="list-style-type: none">• Extends state machine to store sequences• Handles user interface events	<ul style="list-style-type: none">• Does not allow intensive event or sequence processing• User interface events need to be controlled by limiting access to front panel controls

Design Pattern: Producer / Consumer



Design Patterns for CLD Applications

Design Pattern	Advantage	Disadvantage
State machine	<ul style="list-style-type: none">• Handles sequence control	<ul style="list-style-type: none">• Cannot handle storing of sequences• May not be responsive enough to user interface events
Queue based UI Event Handler	<ul style="list-style-type: none">• Extends state machine to store sequences• Handles user interface events	<ul style="list-style-type: none">• Does not allow intensive event or sequence processing• User interface events need to be controlled by limiting access to FP controls
Producer / Consumer with Events	<ul style="list-style-type: none">• Responsive to user interface events• Allows intensive event and sequence processing	<ul style="list-style-type: none">• Does not integrate non-user interface events well

Design your way to success:

1. Design Patterns

2. Timing

3. File IO

4. Error Handling

5. Documentation

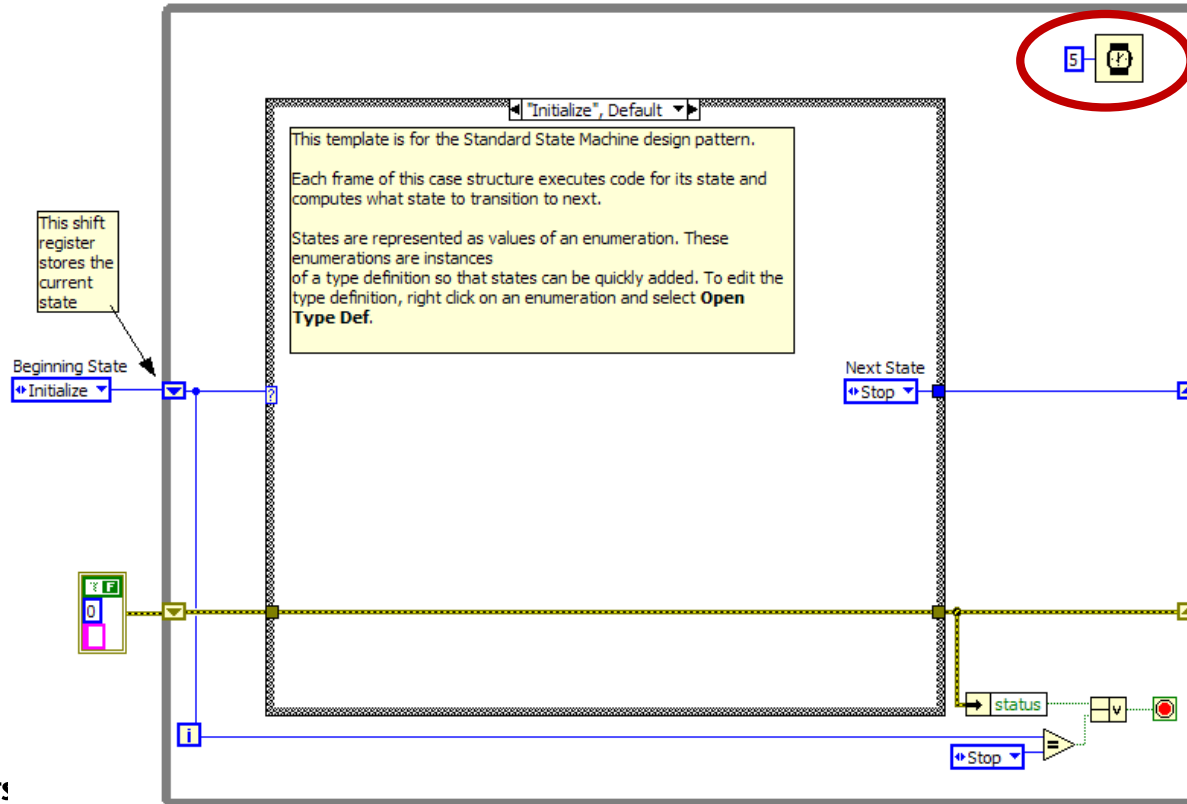
6. Development Style

Timing for CLD Applications

Timing Type	Timing Method	Applications
Execution timing control	Wait (ms) Wait Until Next (ms) multiple	<ul style="list-style-type: none">• Regulate execution of state machine loop to allow other programs to run on the computer• Conserve system resources, prevent “locking up”

- Execution timing control allows us to regulate how quickly a loop executes on the processor
- Beware of data flow

Execution Timing Control



Timing for CLD Applications

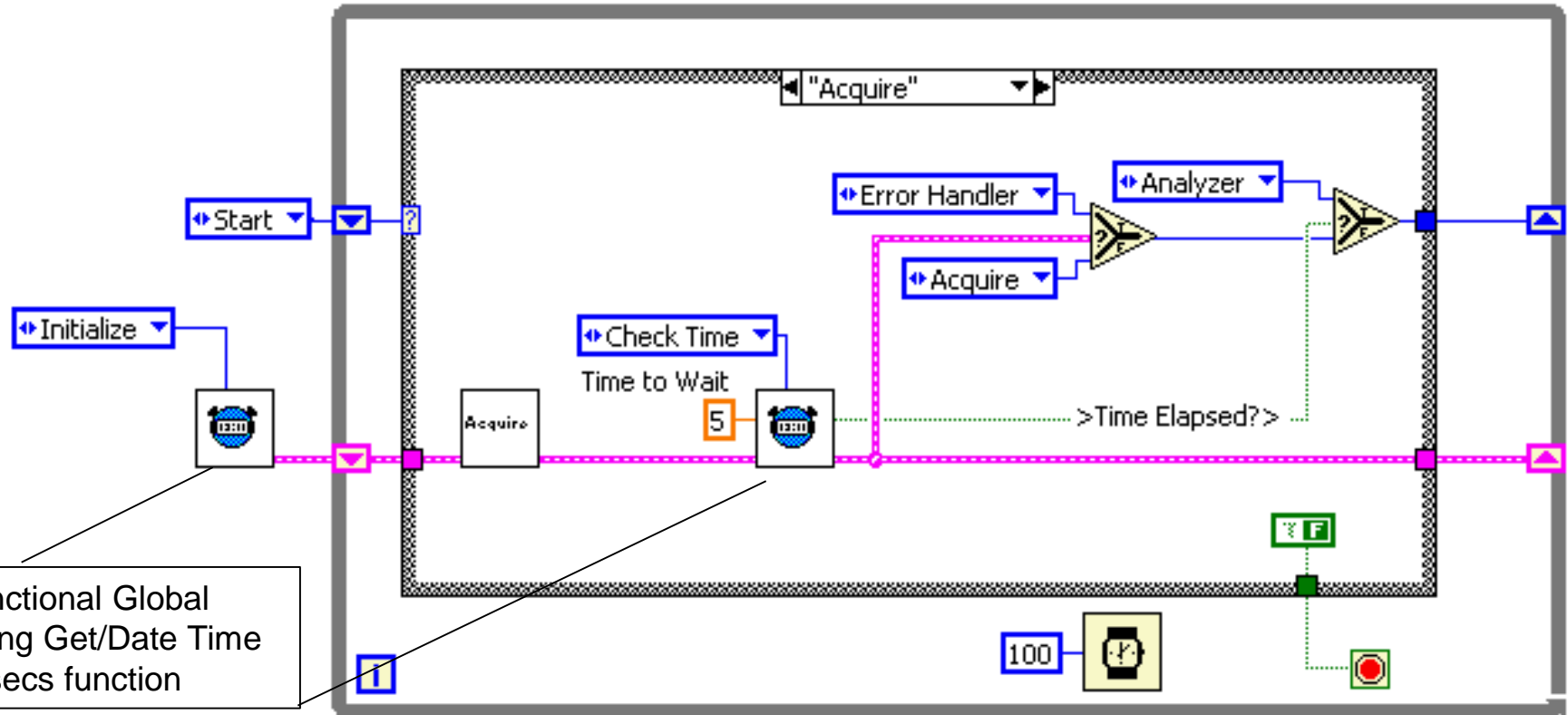
Timing Type	Timing Method	Applications
Software timing control	Get Date/Time in sec. Tick Count	Good general purpose timing method for timing a sequence operation. Encapsulate in functional global or SubVI
	Express Elapsed time VI	Measures Elapsed time with stop / reset control functions. Ready made solution !!

- Software timing control allows us to time a real-world operation to perform at a specific time and/or for a set time period.

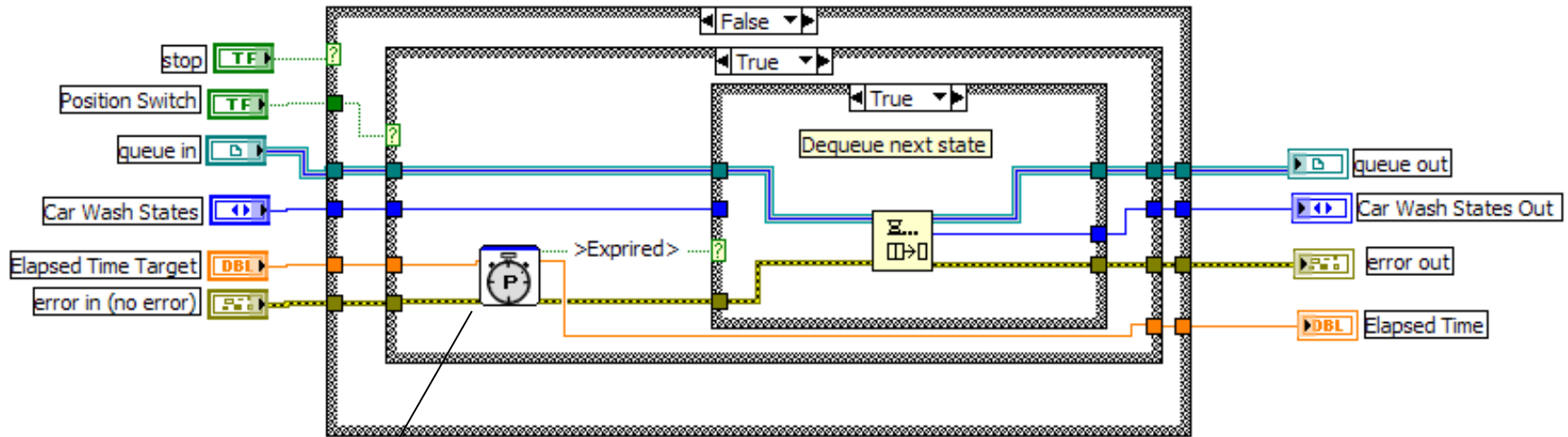
Other useful Software timing methods:

- *Event Case Timeout* – Useful to stop an operation if no user activity is detected
- *Synchronization VI Timeout* - Useful for executing an operation at specific time periods

Software Timing Control Using Functional Global



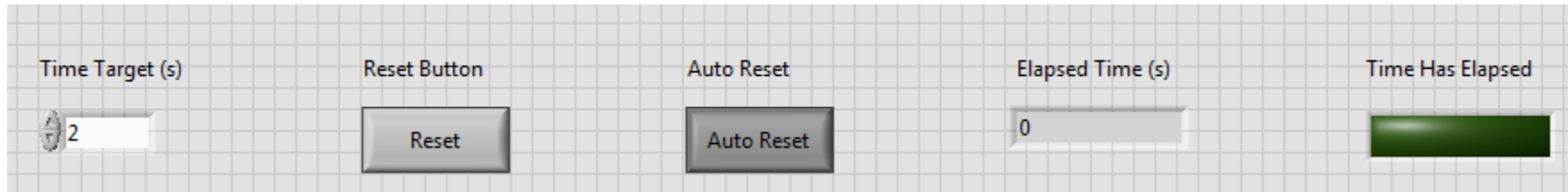
Software Timing Control: Example from the Car Wash sample exam



Elapsed Time
Express VI

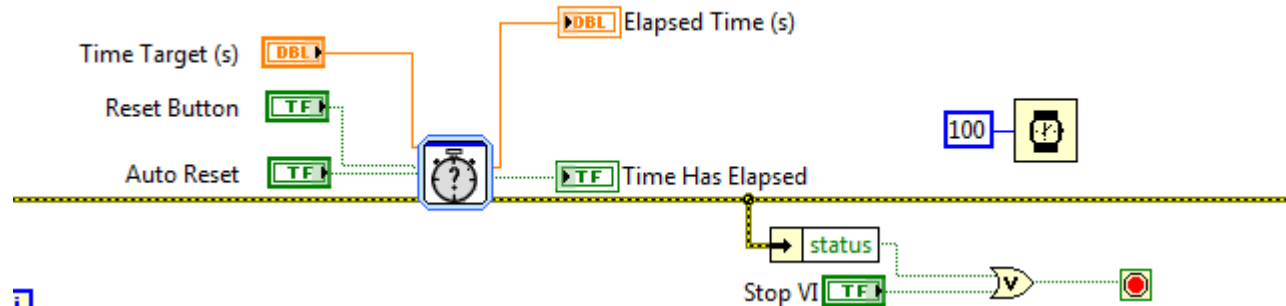
This is a subVI from the Car Wash sample exam

Software Timing Control: Example from Success Package #1

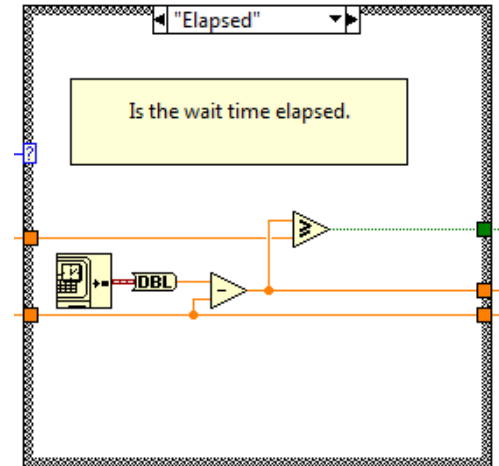
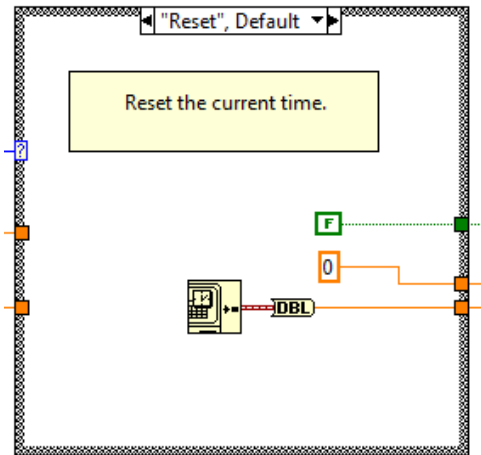
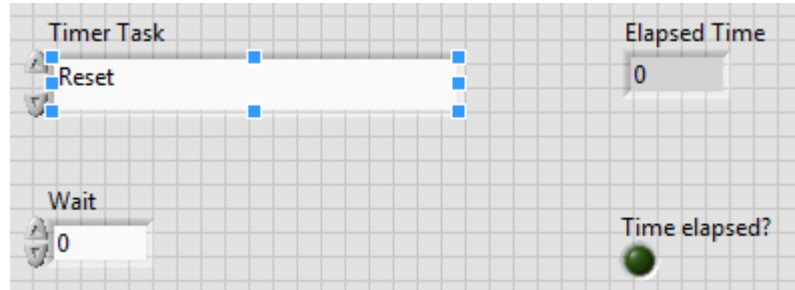


Front panel controls & indicators provided

You program to meet the spec:



Software Timing Control: Example from Success Package #3



Design your way to success:

1. Design Patterns
2. Timing
- 3. File IO**
4. Error Handling
5. Documentation
6. Development Style

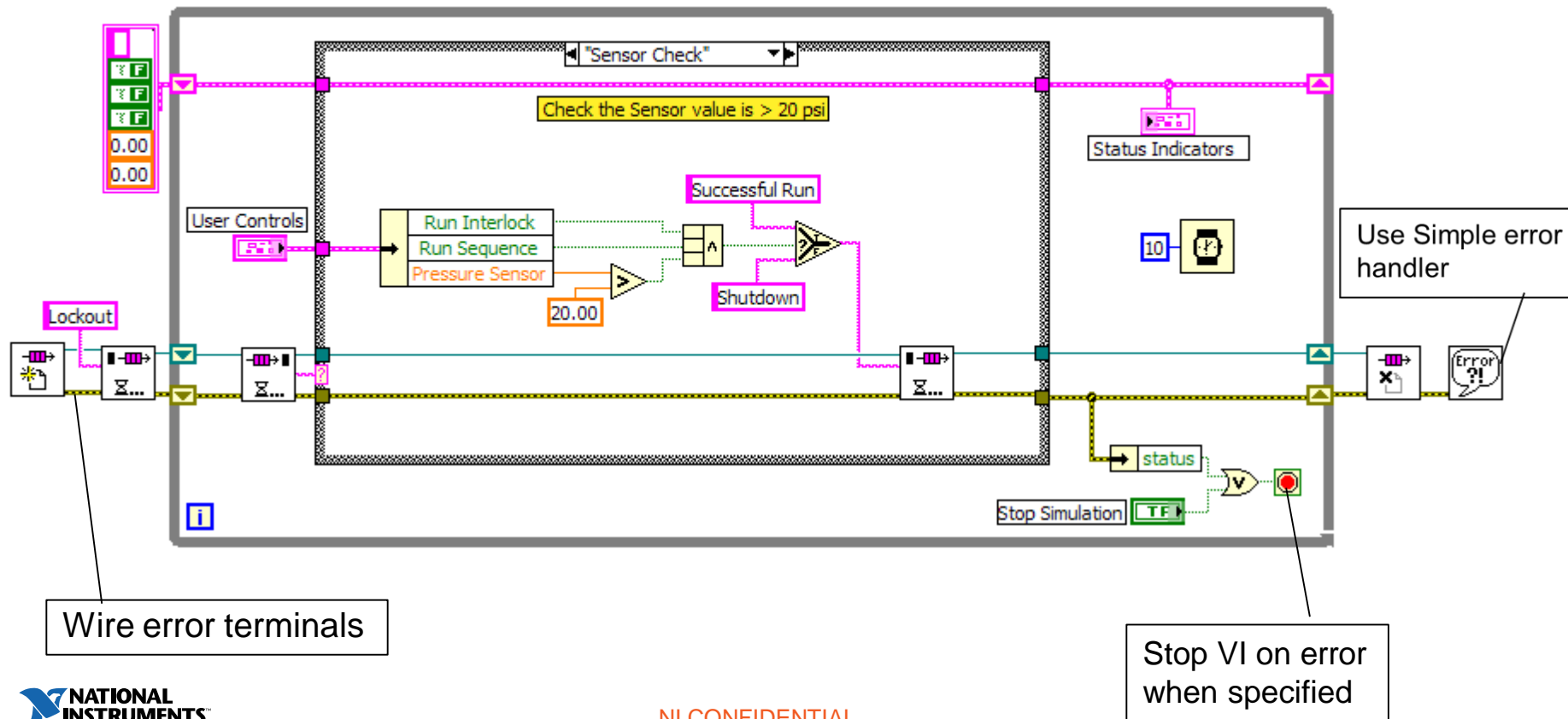
Design your way to success:

1. Design Patterns
2. Timing
3. File IO

4. Error Handling

5. Documentation
6. Development Style



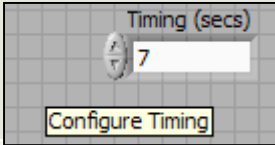
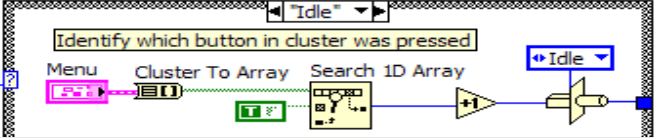
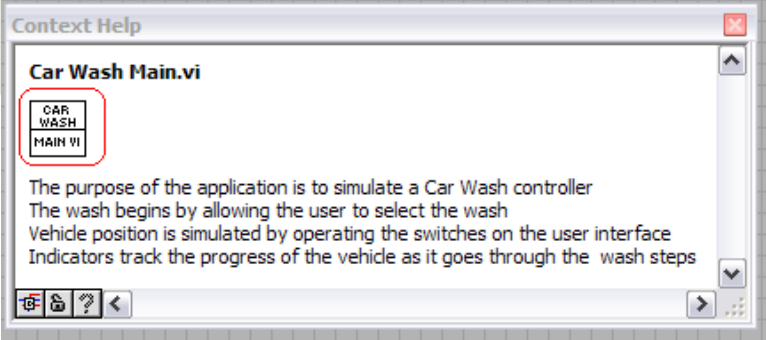
Error Handling in CLD Applications



Design your way to success:

1. Design Patterns
2. Timing Methods
3. File IO
4. Error Handling
- 5. Documentation**
6. Development Style

Documenting CLD Applications

Documentation	Example
Label wires to identify their use	
Label constants	
Description and tool tips for UI controls	
Block diagram comments	
VI / SubVI Properties » Documentation	



Tip: Brevity !!

Design your way to success:

1. Design Patterns
2. Timing Methods
3. File IO
4. Error Handling
5. Documentation

6. Development Style

CLD Development Style

Small things that make a Big Difference !!

- Avoid the use of local variables unless updating a control
- Use property nodes to modify control attributes and control values but not indicator values (hardwire indicators)
- Typedefine reused enums and data structures
- Close references if opened explicitly
- Avoid data coercion and default tunnels
- Avoid copies of code - Develop SubVI if code is reused
- Create readable block diagram
 - Avoid unnecessary bends, overlapping objects, wires

CLD Exam Logistics

1. CLD Exam Process
2. CLD Exam Grading

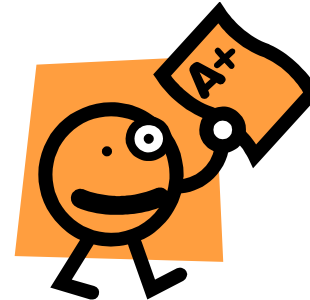
CLD Exam Process

- We provide:
 - A PC with Current Version of LabVIEW
 - Exam packet with:
 - Printed Application Development Specification
 - USB with:
 - VI template establishing front panel (Controls and Indicators)
 - IO file if required
- Test parameters/limitations:
 - No external resources permitted (including no internet)
 - May use any resources that ship with LabVIEW Professional Development
 - LabVIEW documentation, Templates, Express VIs and Example Finder may be used
 - Time allowed for exam: 4 hours
- **Only work saved to provided USB will be graded**

CLD Exam Grading Criteria

Grading Criteria	Points on Exam	Percentage
Functionality	15	37.5%
Style	15	37.5%
Documentation	10	25%
Total Points	40	100%

Passing grade is 70 % or higher



How the CLD is graded

- **Style & Documentation:**

- VI Analyzer does objective count of occurrences in a category
- Thresholds set, category fails if over threshold
- Documentation checks methods (Tools & locations), not content

- **Functionality:**

- **Demonstrated Functionality:** Move controls & watch indicators
 - Points allotted into categories, then sub-divided further
 - Demonstrates implementation of required logic
 - We don't "fix" Vis that are "close" -- Keeps things objective
 - Reflects what a client would want to see in a demo
 - Documentation or complex coding ≠ functionality
 - Therefore: Establish basic functionality & add features modularly

CLD: Exam Tips, Completion, and the CLD-R

1. Exam Tips: Preparation & Exam Taking
2. When the exam is over
3. CLD-R

Tips for Exam Preparation

- **Test yourself:**
 - Time yourself through the sample exams
 - Evaluate your solutions using the Exam Preparation Guide and the evaluation criteria given at the end of the guide
- **Do better:**
 - Incorporate best practices in all your LabVIEW projects
 - Use the generic sequencer model to develop new scenarios
 - Improve your coding speed
 - Find optimal ways to solve problems in LabVIEW
 - Use Success Package to minimize surprises on exam day

Tips during Exam

- Take a few minutes to plan your application
 - Decide on the best main VI architecture for the application
 - Decide which modules / SubVIs you will need to create
 - Decide on the timing method
- Focus on getting the core functionality working
 - Add functionality incrementally, Verify as you go along
- Document the VI and SubVIs as you go
- Pace yourself...Time flies when you're having fun!!
- Towards the end, clean up block diagram
 - And do a last functionality verification! Re-insert USB
- Many more tips in the CLD Success Package!

What happens after the exam?

- Exam proctor sends exam USB to NI for evaluation
- Candidates receive results within 4 weeks
- Successful? Receive CLD shirt, logos and certificate by mail
- CLDs must recertify every three years
 - take the CLD-R exam,
 - re-take the functional exam,
 - or accumulate points.

http://download.ni.com/evaluation/certification/recertification_by_points/recertification_by_points.pdf

- or better still: certify as CLA or/and CLED
- If not successful on CLD: Eval sheet provides feedback
 - Re-attempt the exam when prepared

CLD Recertification Exam

- LabVIEW version: 2009
- 40 multiple choice questions, 1 hour duration,
- Passing grade: 70%
- Offered @ Pearson Vue
- Recertification policy and process on:
<http://zone.ni.com/devzone/cda/tut/p/id/9605>
- Preparation materials – prep guide & sample exams on:
http://www.ni.com/gate/gb/GB_CUSTEDCLDREXAMPREP/US

CLD Recertification Exam: Strategy

- Practice Sample Exam.
 - Timed !
 - Review answers, research gaps, use forum for assistance.
 - Modify conditions of questions, How would modifications change the behavior and answer.
- Repeat with Second Sample Exam

CLD Recertification Exam Topics

#	Exam Topic	Percent Coverage
1.	Architecture / Project	10.0%
2.	Events	10.0%
3.	Error handling	10.0%
4.	Debugging	12.5%
5.	Performance	10.0%
6.	Timing	10.0%
7.	Recursion / Reentrancy	5.0%
8.	Testing	5.0%
9.	Calling external code	2.5%
10.	Shared variables	2.5%
11.	VI server	10.0%
12.	File IO	7.5%
13.	New features	5.0%

This concludes the presentation

Thanks for attending the CLD / CLD-R prep course

Best of Luck on your CLD / CLD-R exam !!!

- Email questions / comments to certification@ni.com
- CLD Prep resources at www.ni.com/cld-prep
- For a previously recorded version of this presentation:
<https://ni.adobeconnect.com/p16xunfj1zb/>

Stay Connected During and After NIDays



ni.com/niweekcommunity



facebook.com/NationalInstruments



twitter.com/niglobal



youtube.com/nationalinstruments

A close-up photograph of a piece of lined paper. The word "Questions?" is written in a large, bold, cursive script using a black marker. Below the word, a single, long, curved horizontal line is drawn with the same marker. The tip of the marker is visible in the lower right corner of the frame.

Questions?

Thank you for your attention

Corrado Degl'Incerti Tocci



The logo for NIDays Engineer Next is centered on a blue background with diagonal stripes. It features the text "NIDays" in white inside a white rectangular box, followed by the words "ENGINEER" and "NEXT" in large, bold, white capital letters. A yellow graphic element, resembling a stylized 'N' or a folded ribbon, is positioned between "ENGINEER" and "NEXT".

NIDays
ENGINEER
NEXT