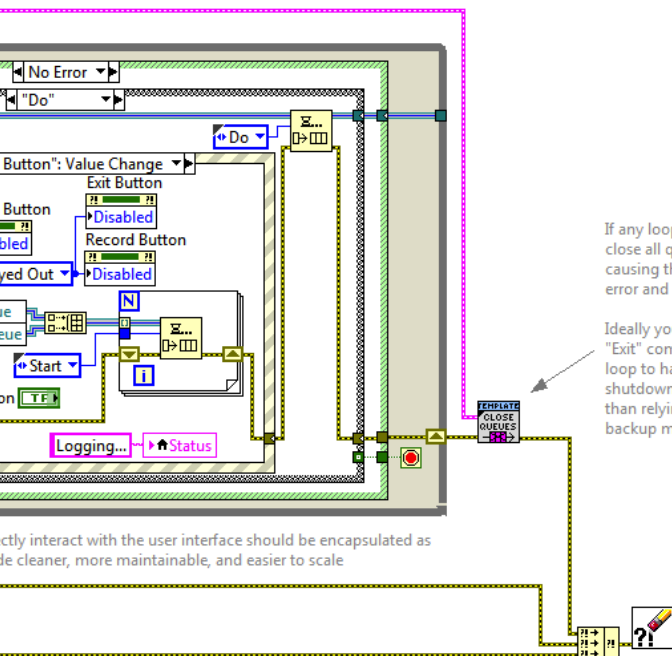


## Template

different rates (i.e. multiple acquisition rates, acquire at one



ctly interact with the user interface should be encapsulated as  
de cleaner, more maintainable, and easier to scale

the data display in it's own loop ensures that drawing the chart  
not interfere with or slow down the acquisition and logging. The  
me wait on notification will produce an error when the notifier  
en released at the end of the program.

# Templates for Common DAQ Applications

# Why Use a Template?

- It's efficient
- Save your thinking for the novel parts of your application

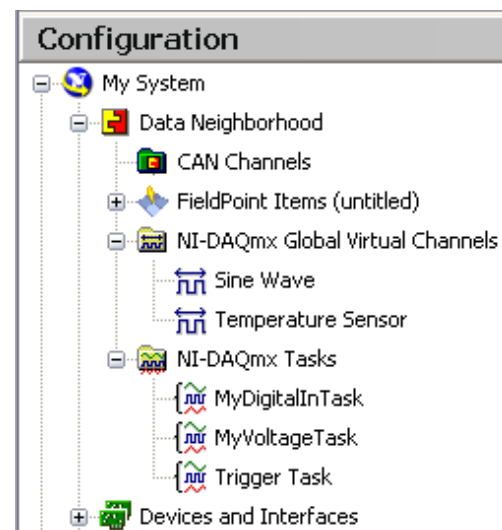
# What We'll Cover

- Preparing to use a template
  - Wrapping hardware APIs
- Some ready to use templates
- Quick Measurement State Machine
- Continuous Acquisition/Monitoring/Logging
- \* Advanced Templates

# Examples

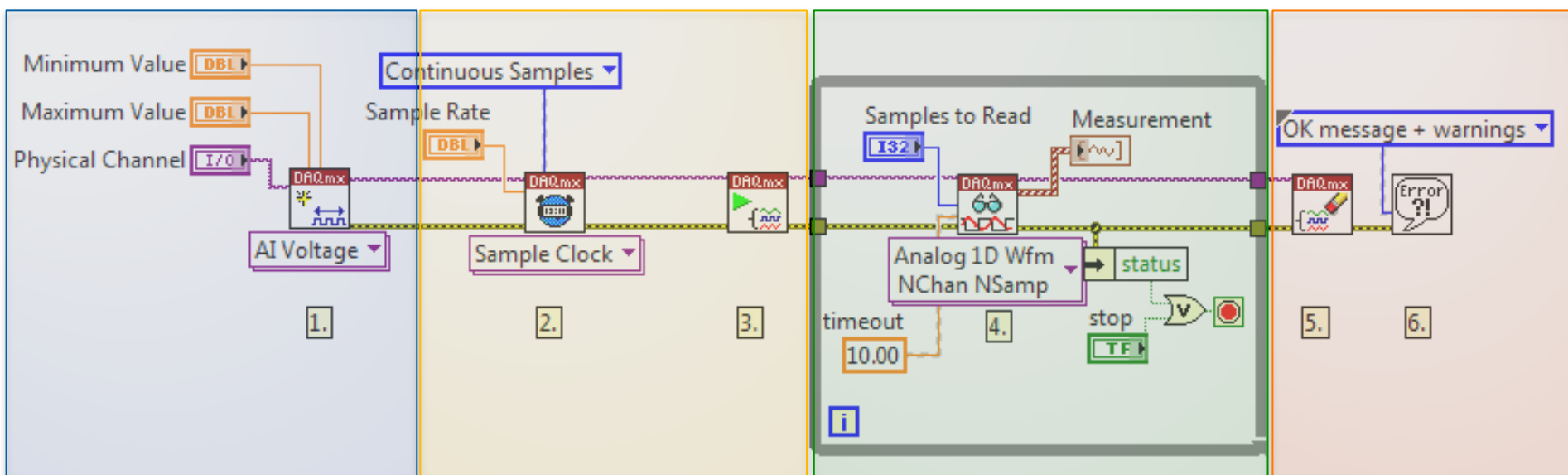


- Use MAX
- DAQmx installs into example finder
- Instrument drivers also come with examples
  - Example Finder
  - Palette
- Other drivers will normally place examples in Example Finder



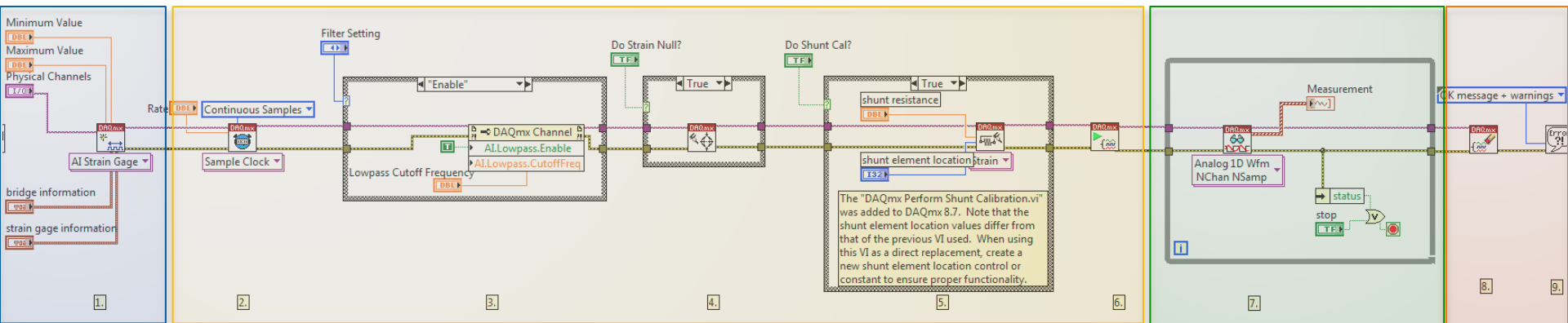
# Common Architecture

Open – Configure – Read/Write – Close



# Common Architecture

Open – Configure – Read/Write – Close



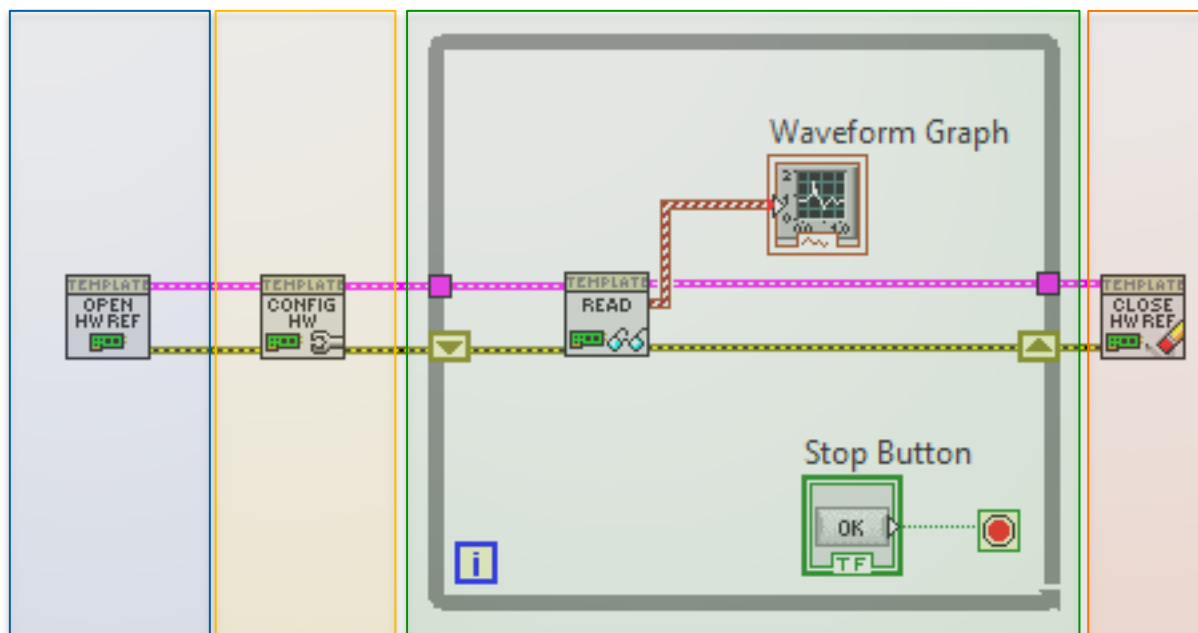
# Apply Good Programming Practices

- Hierarchy
  - Expose details at appropriate levels
- Encapsulation
  - Create reusable, enclosed units (subVIs and libraries)
- Style Guidelines
  - Keep the code quality high (better reusability)
- This same reusable component is used in both templates

# Common Architecture - Wrapper

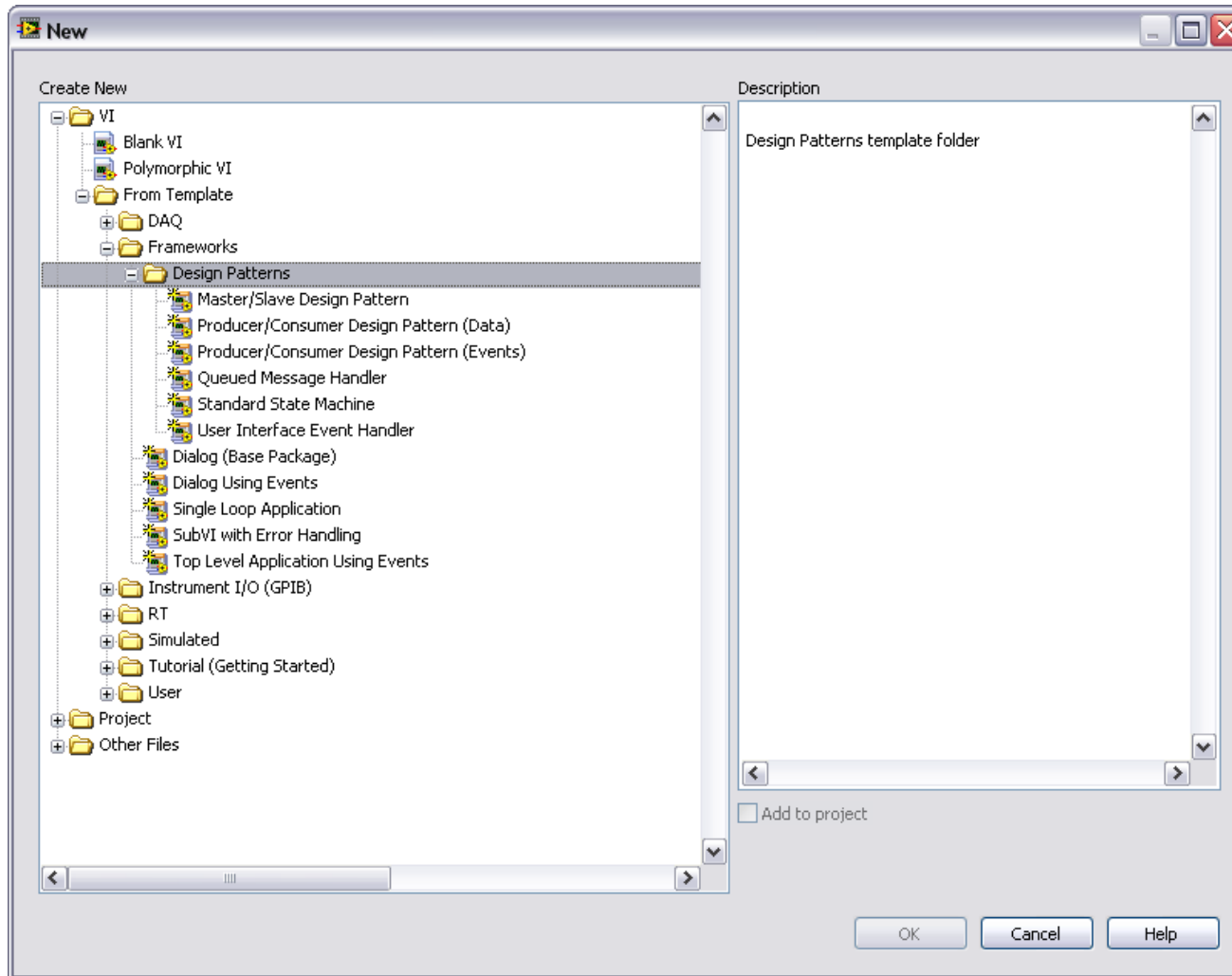


Open – Configure – Read/Write – Close



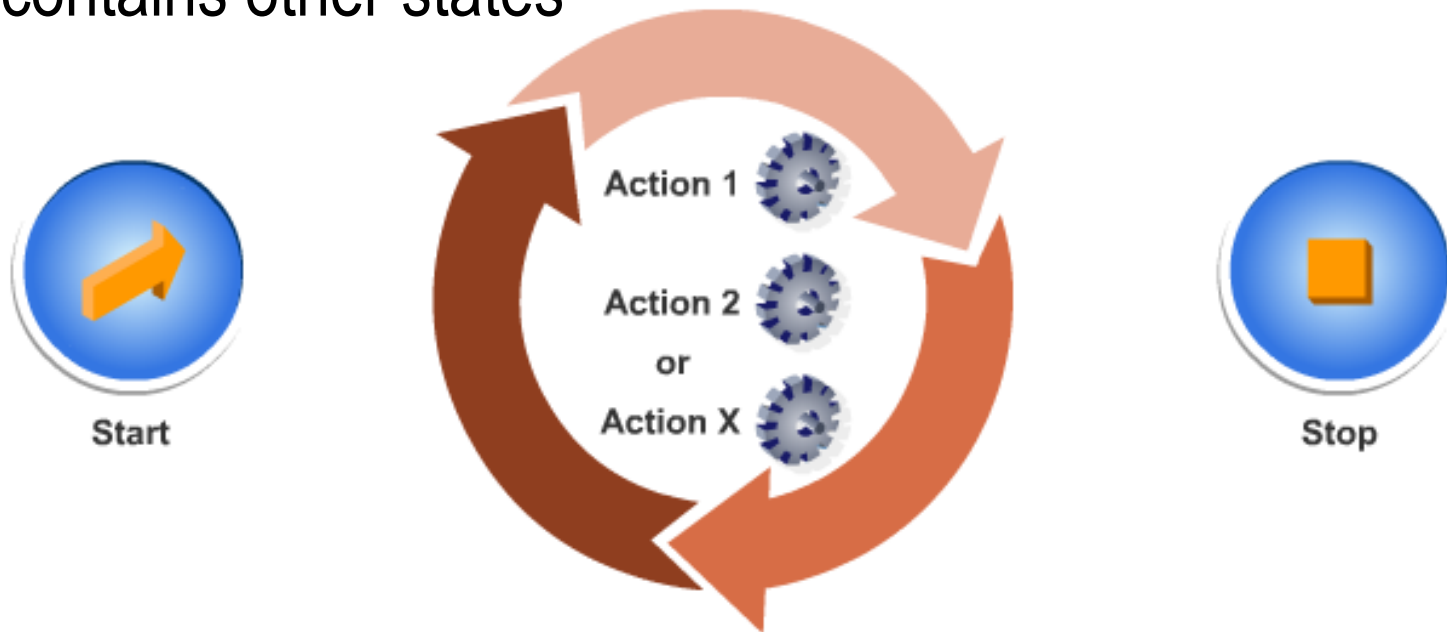


# LabVIEW Templates



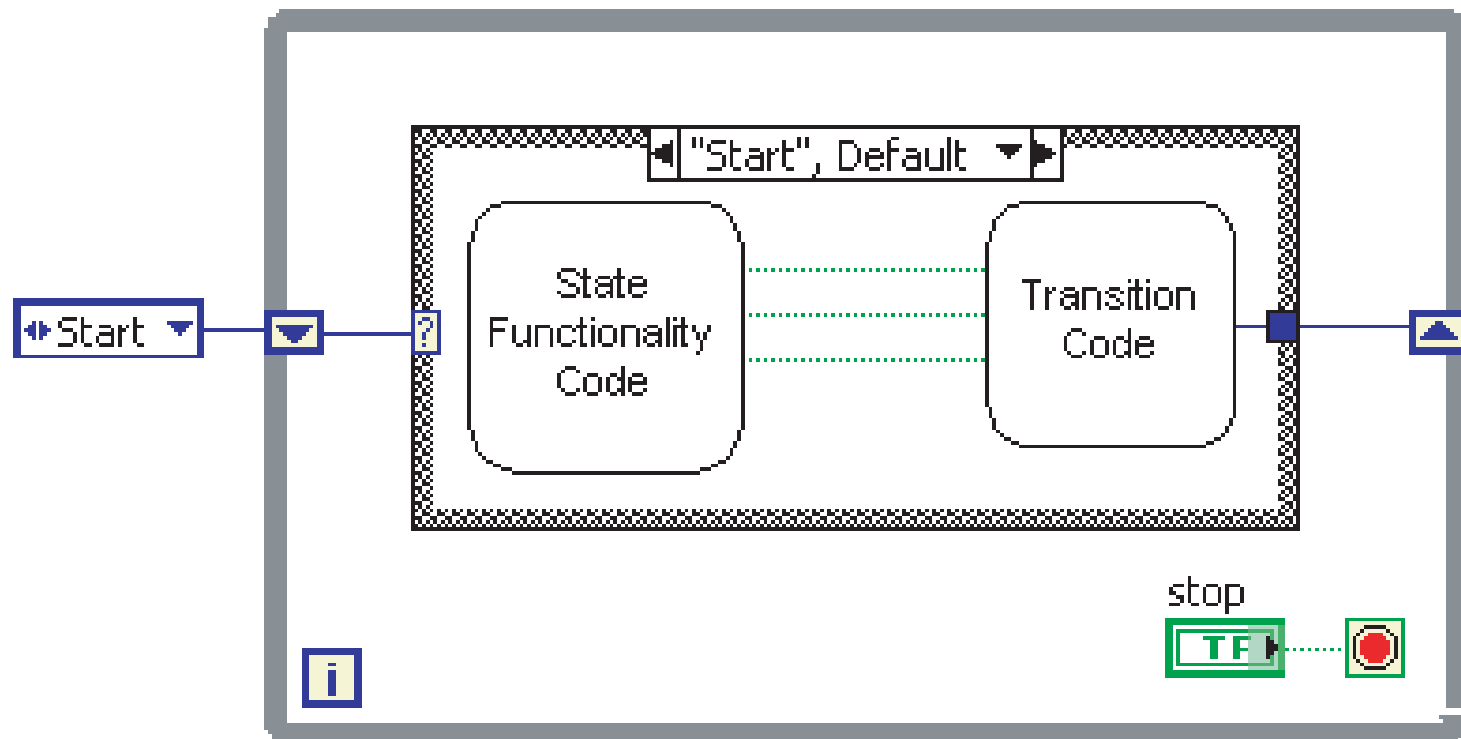
# Single Loop Design Patterns

- State Machine
  - Usually has a start-up and shut-down state, but also contains other states



# Single Loop Design Patterns

- State Machine

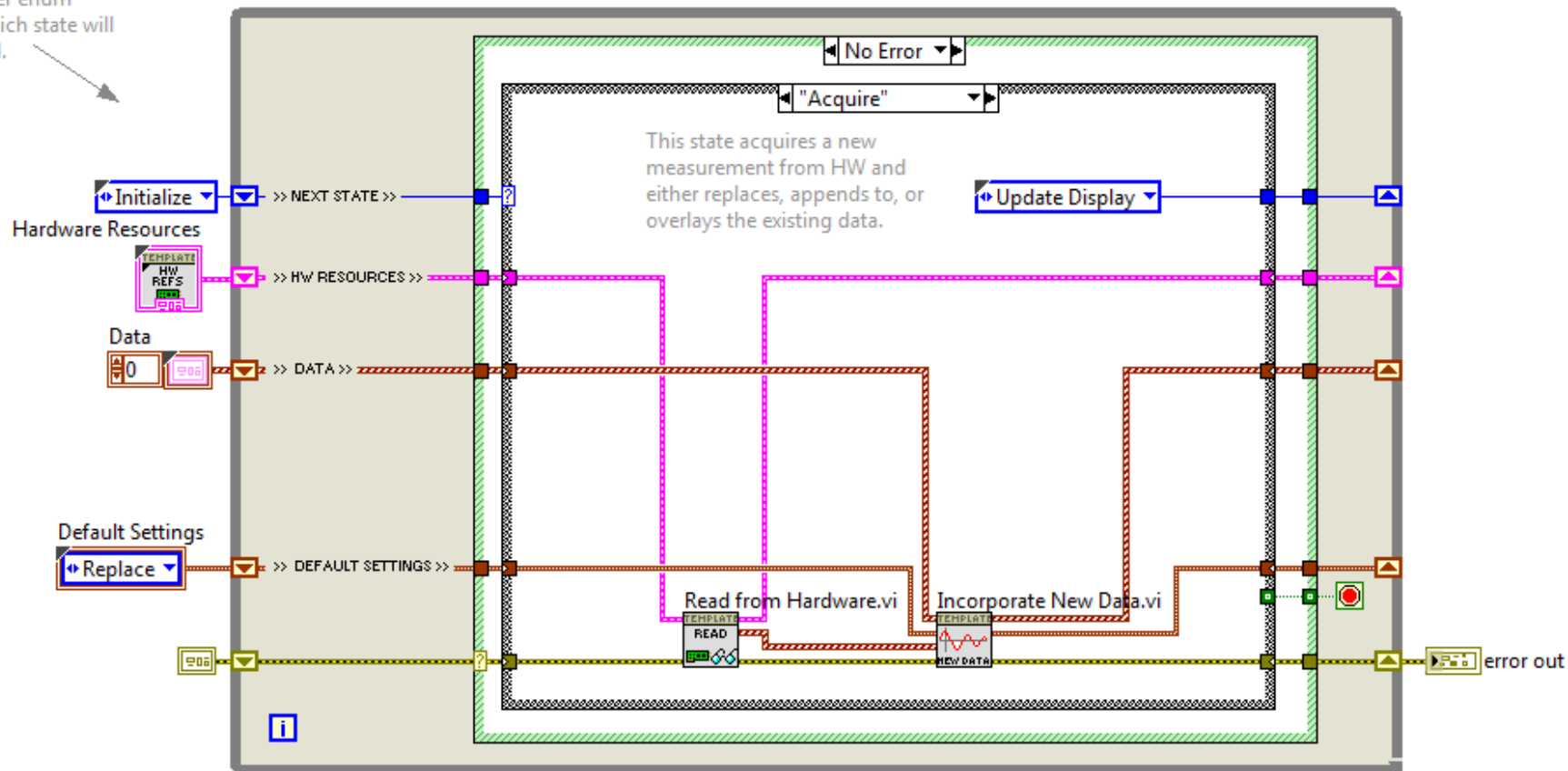


# Quick Measurement State Machine

## Quick Measurement State Machine Template

This template implements a simple state machine appropriate for single-shot acquisition of data from any measurement device.

This TypeDef enum controls which state will be executed.

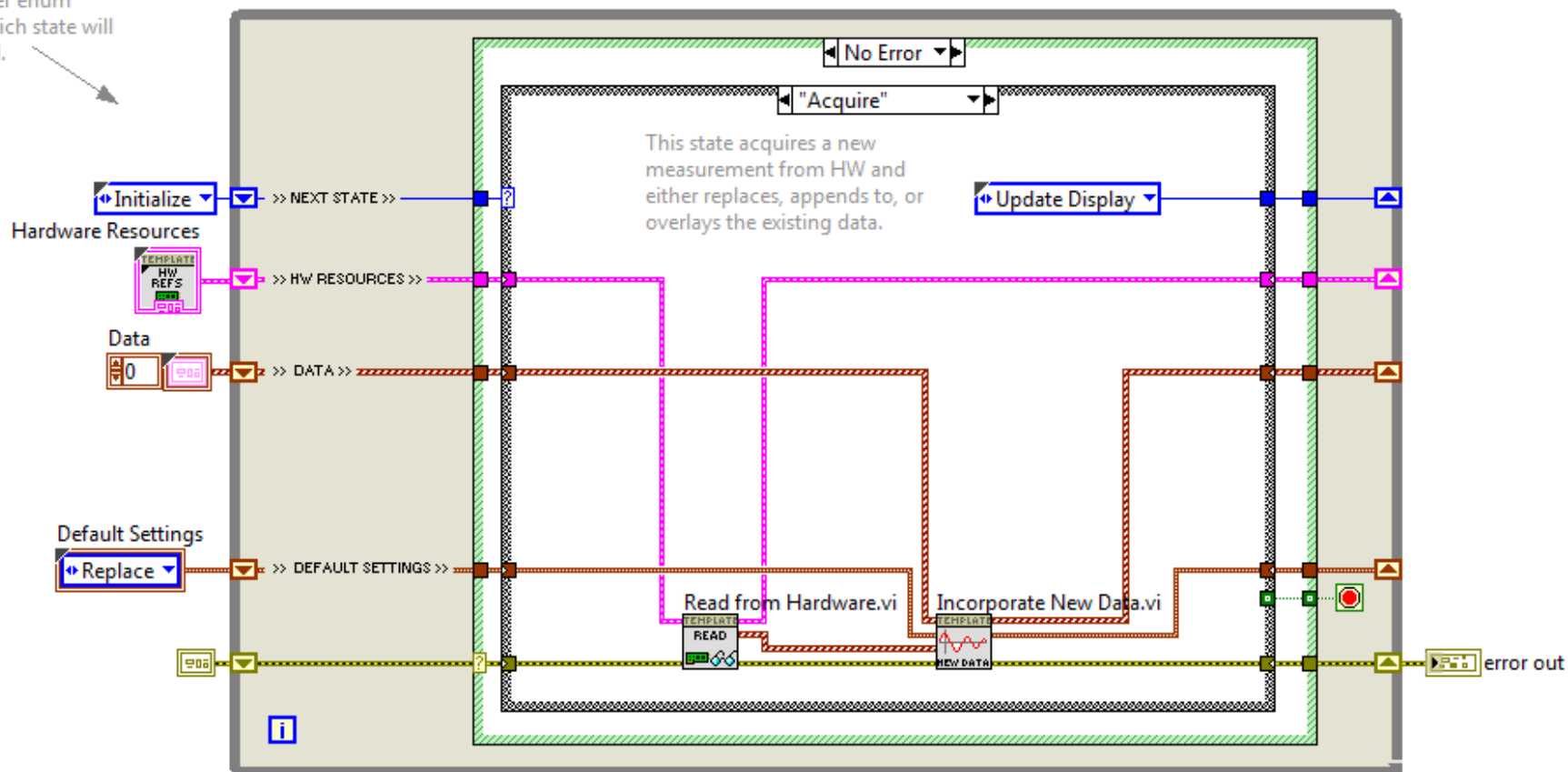


# Look at the Code

## Quick Measurement State Machine Template

This template implements a simple state machine appropriate for single-shot acquisition of data from any measurement device.

This TypeDef enum controls which state will be executed.



# Quick Measurement State Machine

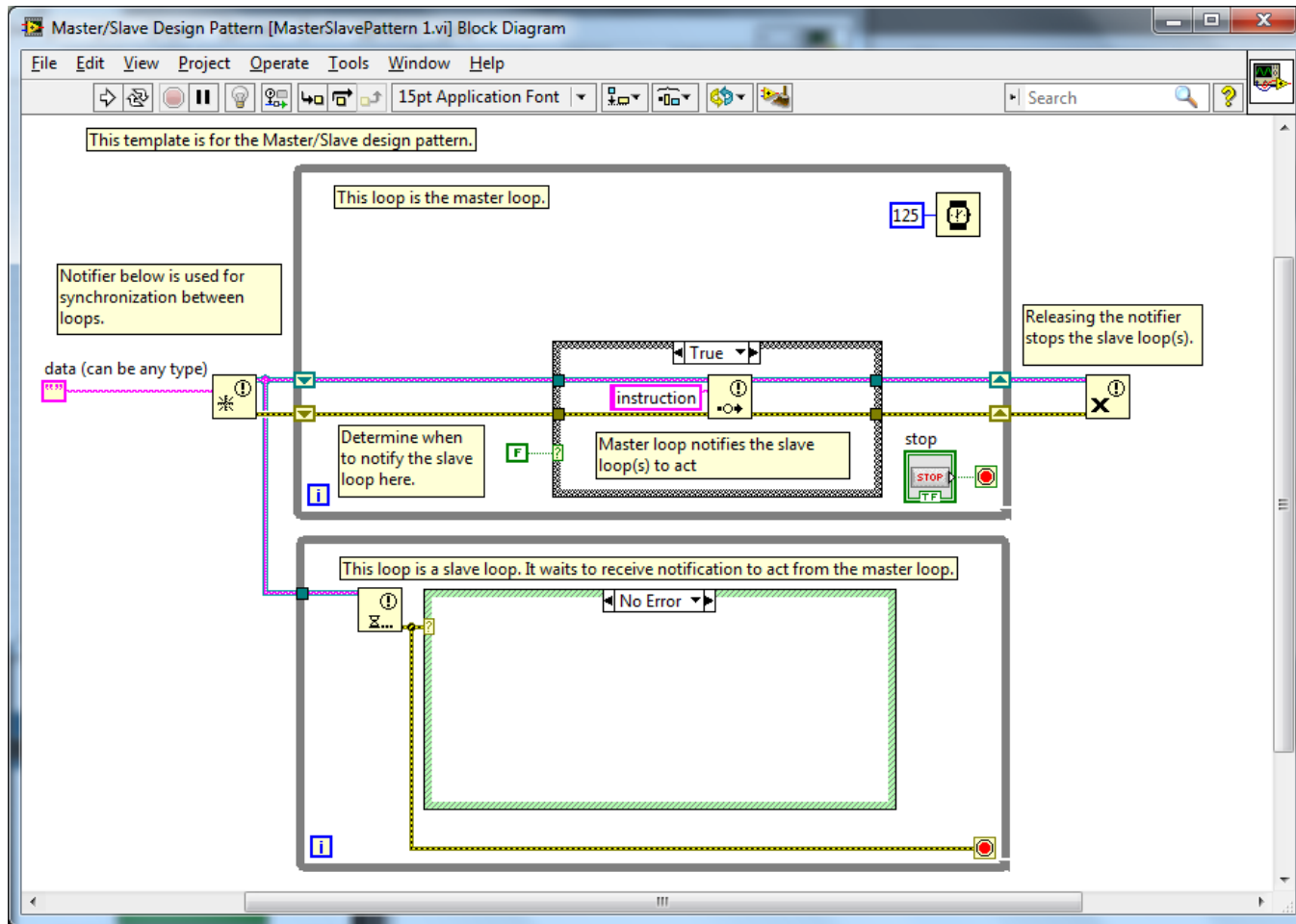
## Good for...

- ... quick measurements
  - i.e. measurements that return data almost immediately
- ... UI-based control
  - i.e. turning motors on/off, generating signals
- ... offline data viewers

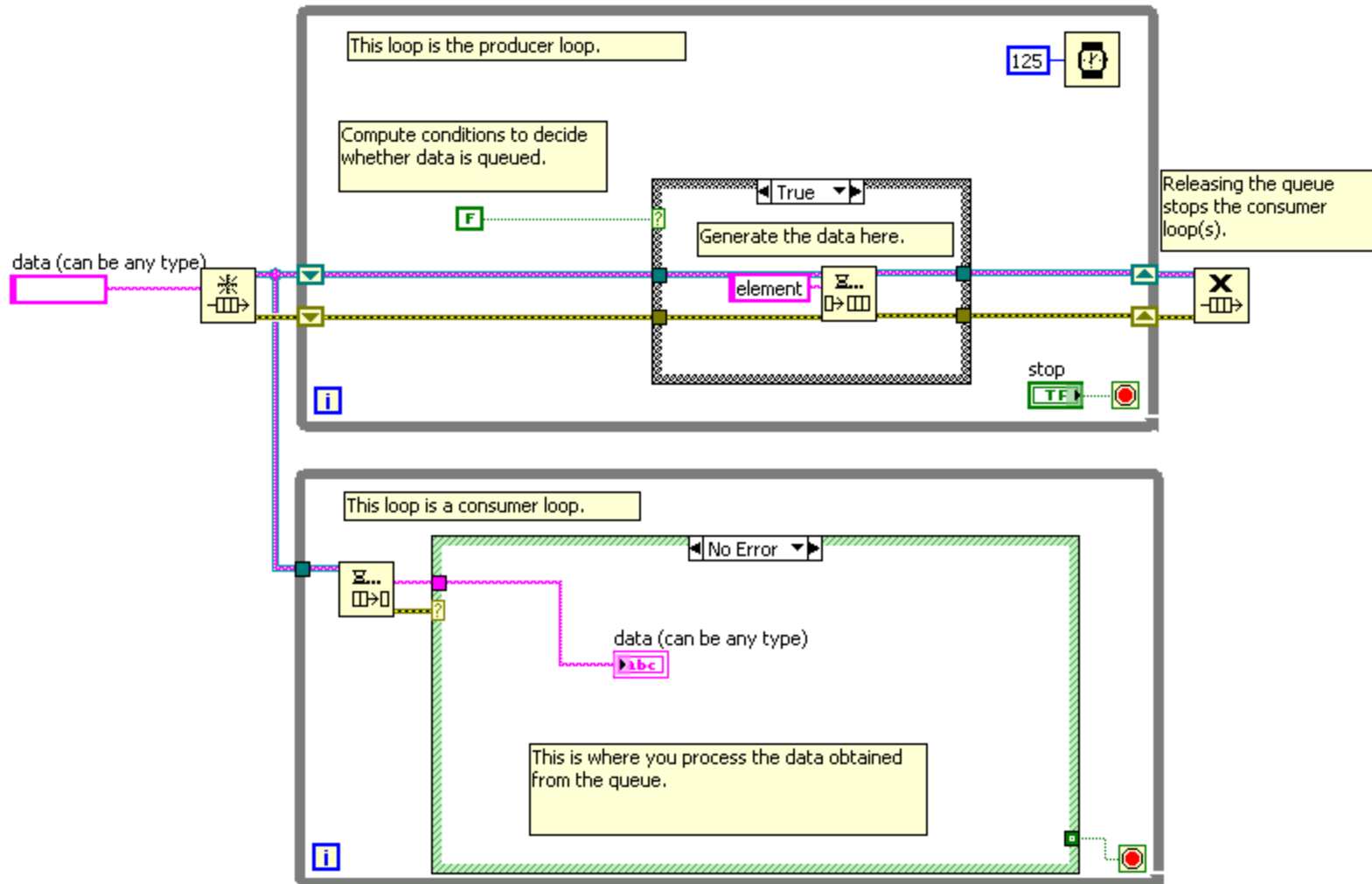
## Not appropriate for...

- ...multi-rate processes
- ...long-running measurements
- ...blocking operations

# Master/Slave Design Pattern

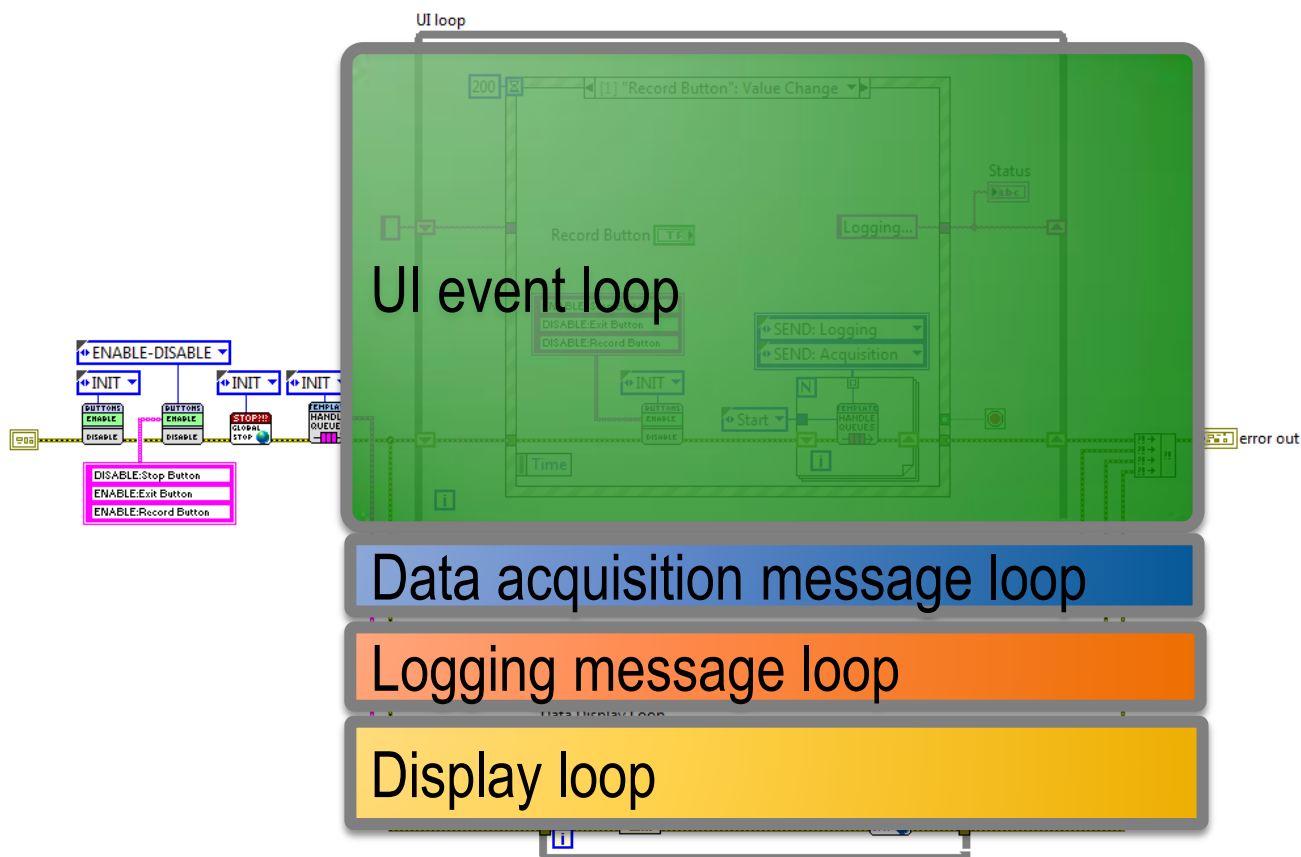


# Producer/Consumer Design Pattern (Data)





# Continuous Acquire



# Continuous Acquire

UI Command Queue >>

UI event loop

Acquisition Command Queue >>

Data acquisition message loop

>> Data Queue,  
Data Notifier

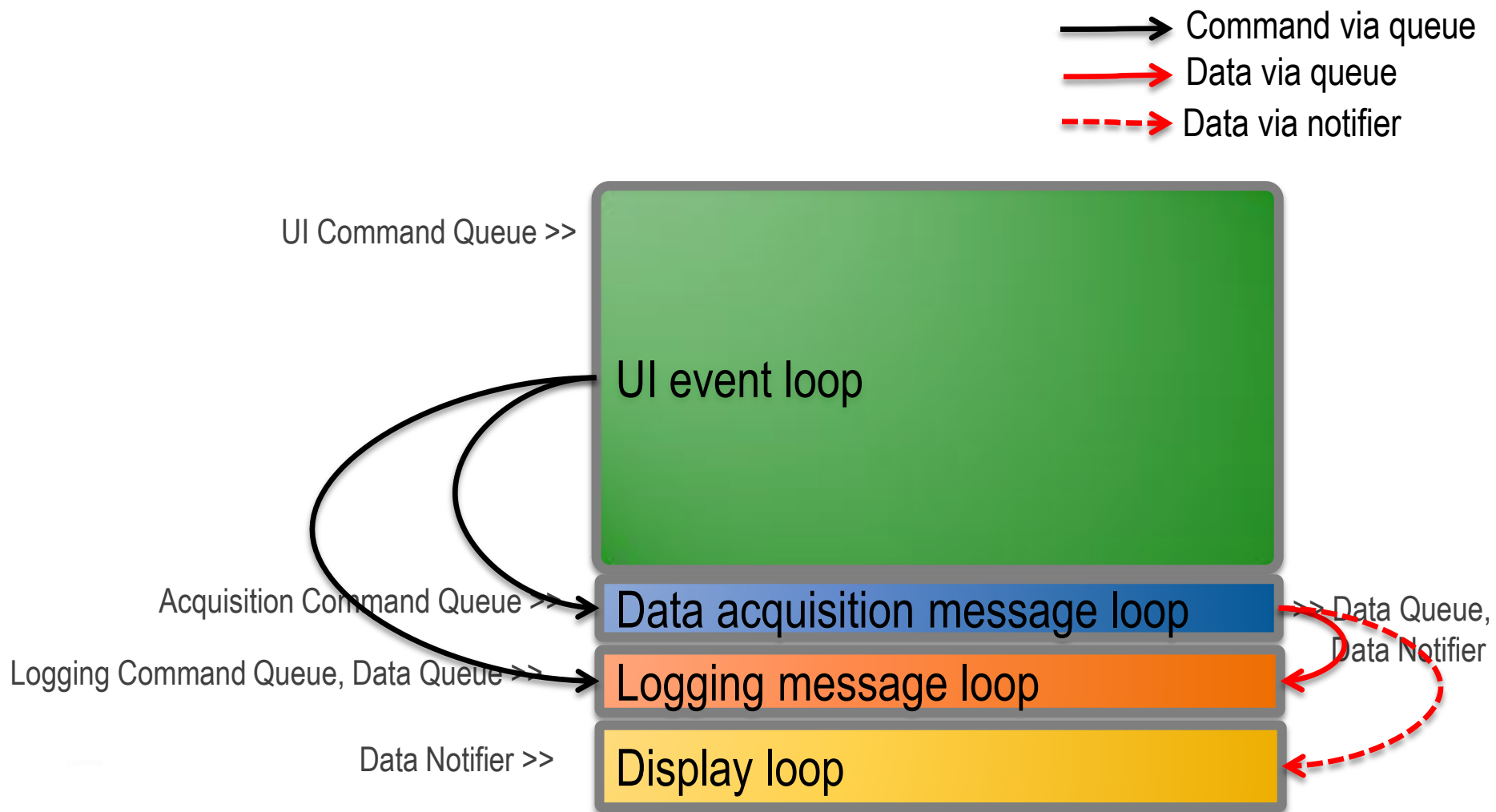
Logging Command Queue, Data Queue >>

Logging message loop

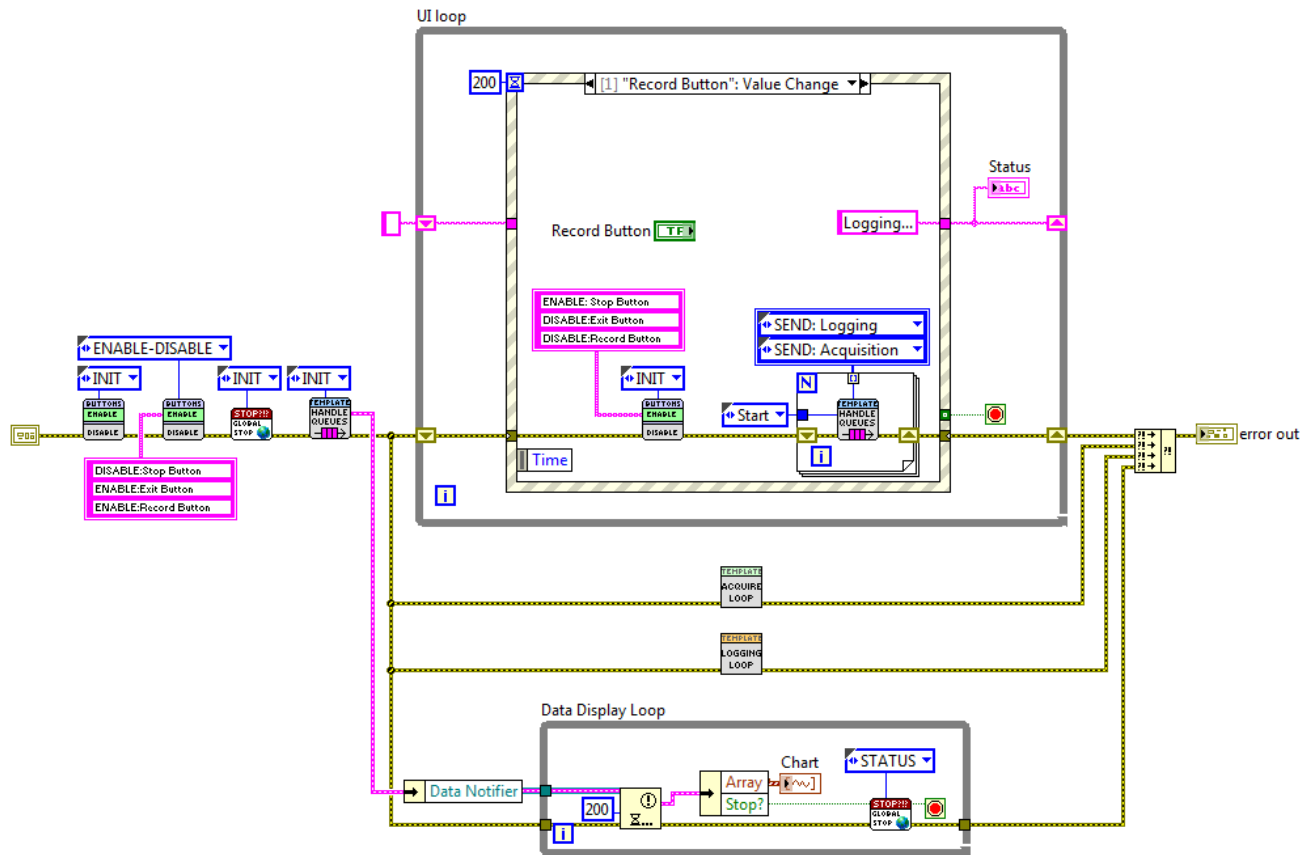
Data Notifier >>

Display loop

# Continuous Acquire



# Look at the code



# Continuous Acquire State Machine

## Good for...

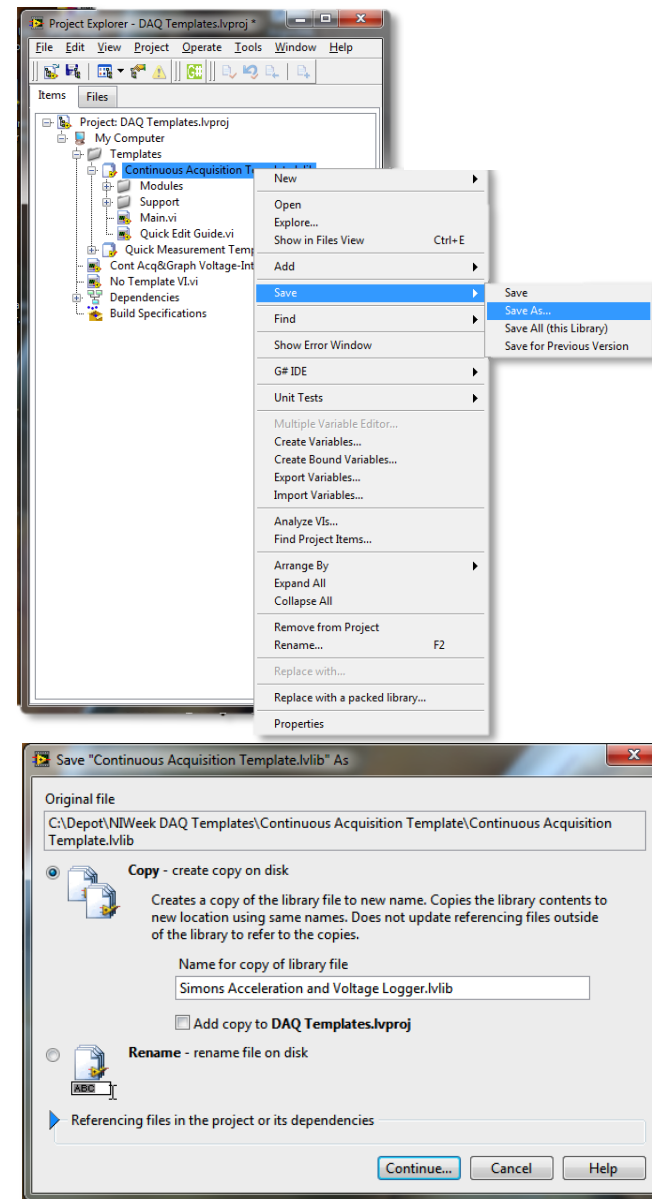
- Efficient UI response
- Long Running measurements
- Processing data continuously

## Not appropriate for...

- Single measurements
- Direct control from UI (turning on/off equipment)

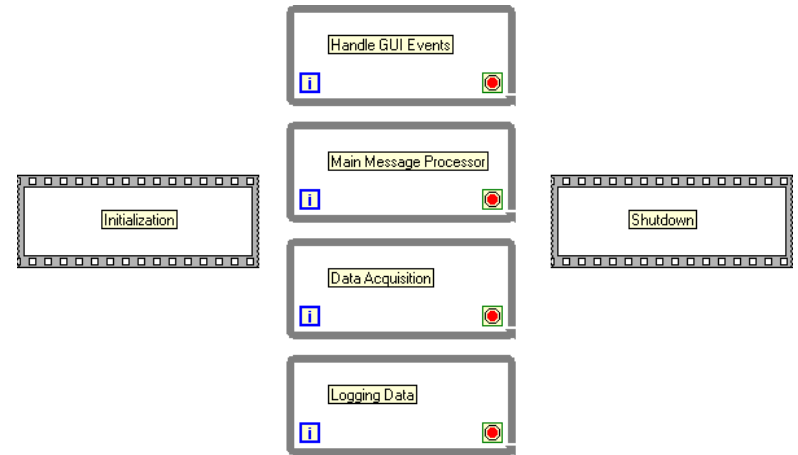
# How to Use

- Right-click on library and select “**Save As...**”
- Leave “**Copy**” selected and give a new name
  - Be wary of 255 character file path limit



# Other Templates

- JKI State Machine =
  - <http://jki.net/state-machine>
- Data Acquisition Reference Design
  - <http://zone.ni.com/devzone/cda/tut/p/id/11805>
- LabVIEW Actor Framework
  - <https://decibel.ni.com/content/docs/DOC-17181>



- *Each of these terms is “Googleable”*

# Thank You! Questions?

- You will get a download link to presentation/demos ;-)

Recommended courses:

- Core 2
- Core 3
- Data Acquisition