# NX24X7: Challenges and Considerations when Designing a Life Cycle Tester

Katya Prince Prince Consulting, LLC Durham, NC katya@princeconsultingllc.com July, 2024

### Introduction

#### Definition:

• System that repeatedly cycles a product over an extended period of time often to device failure.

#### Purpose:

- R&D and product development
- Industry Qualification
- Design Consideration:
  - 24-7 Operation
  - Multi-DUT Testers (Parallel or Batch)

### System Design Assumption

- LabVIEW Application
- PC with Windows
- Communication with NI DAQ, cRIO or Stand-alone equipment

### Example Hardware

#### • DUT:

- Thermoelectric (Peltier) Device
  - Apply voltage/current and will create temperature differential across the device
  - Used for heating or cooling applications

### Example: R&D or Product Development

#### When Used

- New product development
- Design change qualification
- Goals
  - Parallel Execution
  - Flexibility to start-stop each DUT
  - Flexibility to change testing parameters for each DUT
  - Detailed information on each DUT over time

### Example: R&D Hardware Layout



### **Example: Qualification**

#### When Used

• Devices are ready for production, but may need to pass industry (e.g. automotive or aerospace) qualifications for multiple parts.

#### • Goals

- Lot testing for identical devices.
- Statistical data collection on the entire batch of DUTs.

### **Example: Qualification Hardware Layout**



### **Design consideration 24X7**

- Must run with minimal user intervention
- Will be active the majority of the time
- Example of other 24-7 apps
  - Production Line Testing
  - Process Control

### 24X7: Build for WHEN not IF (pt1)

- Power Interruption or Loss
  - Mitigate with UPS on PC and AutoPause/Restart
- Unsafe/critical situations
  - Over Temperature
  - Air Pressure or Water Pressure Loss
  - External Equipment Failure
  - Mitigate with Alarming Pause Testing with user intervention to restart

### 24X7: Build for WHEN not IF (pt2)

#### Communication Loss with Equipment

- Mitigate with automatic retries
- Mitigate with AutoPause/Restart

### • Network Communication Loss or Server Reboot (for Files or Dbase)

• Mitigate with Local Storage and Retries

#### • PC or Application Crash/Reboot. Hard Drive Failure.

- User intervention consider system action logging.
- Create reinstall files and installation instructions.
- Use cRIO for mission critical to safely stop test if PC loses coms.
- Watchdog timers and Master Control Relays.

### 24X7: Hardware Communication Tips

#### AVOID USB

- Build in retries.
  - Some equipment will go into a sleep mode and may need to wake up before responding correctly
- When changing settings, read back values to verify actions
  - This may be a bigger deal with serial drivers that do not reply

### 24X7: Memory Management Tips (pt1)

Try to avoid building/resizing array.
Tip for array update using inplaceness:



• Truncate status display strings

## 24X7: Memory Management Tips (pt2)

#### Close References

#### https://www.ni.com/en/support/documentation/supplemental/13/clo sing-references-in-labview.html

#### • Tools for tracking issues

• Tools-> Profile

Measurement & Automation Explorer		1
Instrumentation	•	
Compare	•	
Merge	•	
Profile		Performance and Memory
Security	•	Show Buffer Allocations VI Metrics
User Name		
Build Application (EXE) from VI Source Control		Find Parallelizable Loops
		Profile Buffer Allocations Trace Execution

### 24X7: File/Data Tips

- Save as you go. Do not wait for testing to be stopped
- Break up files based on size or time
- If networking issues, may want to save local and move to remote
- During design calculate how much data you will be creating
  - May want to separate file into year or month folders
  - May need to reduce data collection either by decreasing scan rate or increasing 'detailed' data collection interval
  - Consider database size
- Don't keep all the data in memory
  - Collect, summarize, log and toss the raw data.

### 24X7: PC Settings

- Disable Auto Updates and Restarts
- Disable File Indexing especially if doing a lot of file IO
- Disable automatic defrag

### 24X7: LabVIEW Licenses

- ALWAYS use EXE
- ALWAYS use deployment licenses (e.g., Vision, TestStand, OPC)

### 24X7: Operation Tips

- Build in Pause/Resume
- Update the UI with Current status
  - for long delays, update a timer
  - No ms wait longer than 1 or 2 seconds long delays decrease application responsiveness => Frustrates user
- Update Test Results as you go
- Periodically save status to file/database

### N: Introduction

In R&D, each DUT are tested in parallel
In Qualification DUTs are batch tested





### N: R&D Parallel DUT Tester



### N: Determine Shared vs Individual Resources

#### Shared Devices

- 32xTC DAQ board
- Chiller
- Other possible shared Resources
  - Communication bus e.g. GPIB, RS485
  - Error Log Files
  - System Log Files
  - Configuration Files

### N: Re-entrancy (pt1)

- Definition
- Use Re-entrancy and dynamic launch to create duplicate 'Test Stages'

### N: Re-entrancy – Example Launcher



### N: Re-entrancy (pt2)

#### VIs should be Reentrant 'all the way down'

- Non-reentrant SubVIs could block action of reentrant caller
- Most device drivers are NOT reentrant. You may need to re-write them
- VI Analyzer has a reentrancy test
- May not want reentrancy in things like Test Start Window

### Nx24x7: SW Packages

- DQMH
- JKI State Machine
- Actor Framework
- TestStand (maybe?)
- Roll your own

### Nx24x7: UI Suggestions

- Allow for summary information and then detailed information
- Use 'pause all'/'resume all'/'stop all'
- Build with extension in mind. Don't let the software be the limiting factor.

### Example UI



### NX24X7: Troubleshooting

Behavior: Application continues to grow

- Possible Reasons: Unbounded array or string. Unclosed references
- Behavior: UI updates and data collection gets slower as more stages are activated.
  - Possible Reasons: Something is non-reentrant.



Life cycle testing is needed by industry, and knowing the core considerations will help you address the myriad of challenges when building such a system.