Refactor vs. Rewrite

Brian Powell

#OurGiantsAreFemale

Elsie Shutt

"What it turned into was a feeling of mission in providing work for women who were talented and did good work and couldn't get part-time jobs."

- Early 1950's, Programmer at Aberdeen Proving Ground and Raytheon
- Forced by law to quit in 1957 after becoming pregnant
- Founded Computations Incorporated in 1957—primarily female freelance software developers

Who Am I?

- CEO of Stravaro LLC
- LabVIEW Champion
- Certified LabVIEW Architect
- Certified Professional Instructor
- LabVIEW R&D 20+ years, NI 26 years
- B.A. Computer Sciences

Passionate about

- team culture
- organizational structure
- people management
- software engineering processes

Today’s Presentation

- You’re given an old LabVIEW application to update
- How would you decide whether to refactor or rewrite?

- I’ll go through a real-world example
- I made some good decisions—and some bad ones

- What would you have done differently?

Part 1: I’ll lay out the problem

Part 2: Discussion

Part 3: The path(s) I chose

The Application

Counter/timer application

- A component of a larger system
  - Communicates with an HMI and PLCs
- Detect a sequence of digital input events, and generate digital outputs relative to those outputs
- Three PXI-6602 boards
  - ~20 timed input signals each + 3 static digital inputs
  - ~5 timed output pulses
  - ~5 static digital outputs
  - Also measure edge-to-edge timing
- Examples:
  - Generate a pulse 5us after the rising edge of an input
  - Measure the time between rising edges of signal A and signal B
  - Generate an output clock at frequency N, then shift to frequency N/4 while signal C is asserted

Note: The current system is not broken!

Some Problems

~20-year-old LabVIEW + DAQ application

- LabVIEW 5.0 + Traditional NI-DAQ
- NetDDE for communications

- Upgraded once to LabVIEW 8.2 + DataSocket for communications
- Windows XP

- Documentation was written last year
  - Thorough but dense
  - Focused more on system behavior, not big picture
- No test procedures

The Contract

FIXED PRICE CONTRACT

Primary goals:

- Update system so that it runs on Windows 10
- Write acceptance test procedures

Secondary goals, improve maintainability:

- Update to newer NI hardware
- Update to latest LabVIEW and DAQ

New system should be functionally equivalent to old system

The Code

Outdated LabVIEW 5.0 programming style

- Loop around case structure to set up digital lines
  - Allows for early exit upon error
  - 67 nearly identical cases, one for each input
- Sort of an overall event handler to set mode/state
  - Top-level case structure for mode
  - Main modes have a big while loop surrounding another case structure

Lots and lots of duplicated code

- Two nearly identical PXI chassis
  - Nearly all VIs duplicated with “-1.vi” or “-2.vi”
- Two of the modes were nearly identical
  - Code was duplicated into different cases/loops

Refactor or Rewrite?

- What would you do?
- What questions should I have asked?

My first thought

Refactor!

- Probably lots of undocumented behaviors that need to be retained (e.g., “magic” delays)
- Build tests for old system
- Incrementally modify code, ensure tests continue to pass

Plenty of success

- DAQmx API is clean compared to Traditional DAQ
- Lots of code cleanup/elimination of redundancy
  - Counter setup: 67 cases became a few VIs wired together
  - Error case structures
- A few simple dialogs changed from polling to event structures

How that plan failed (1 of 4)

Building tests for old system was much harder than expected

- This application was dependent on undocumented behavior in the “rest of the system”, such as PLCs and sensor signals. This was identified as an early risk.
- Pre-pandemic mitigation plan: Go visit the site and capture signals to playback – THIS NEVER HAPPENED BECAUSE OF PANDEMIC

Weekly calls with Subject Matter Experts

- Experts in the theory of the system
- Lack of ground truth (operators rarely attended calls)

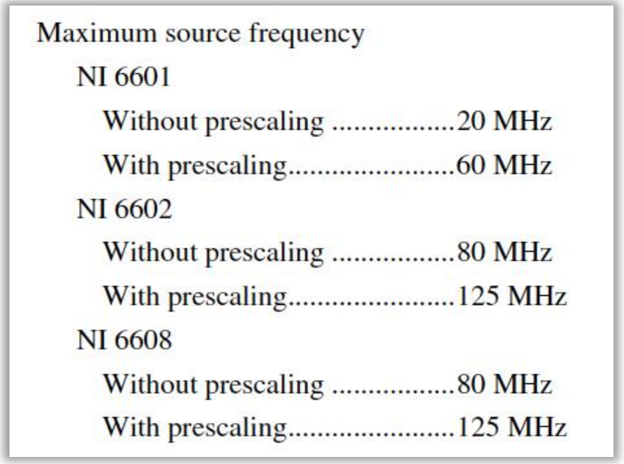
Customer hardware did not exactly behave as documented

- Had to investigate what was “right”

How that plan failed (2 of 4)

Issues with NI hardware, software, and support

- New hardware (PXIe-6614 vs. PXI-6602) was supposed to be compatible, but better/faster
  - Truth: better/faster, but not entirely compatible
  - Datasheets of 6602 and 6614 prevent comparison



Maximum source frequency

NI 6601

Without prescaling ....20 MHz

With prescaling.......................60 MHz

NI 6602

Without prescaling ....80 MHz

With prescaling.......................125 MHz

NI 6608

Without prescaling ....80 MHz

With prescaling.......................125 MHz

6602 data sheet

80 MHz

Had to expand scope to rewrite HMI, which assumed 80 MHz timebase.

100 MHz

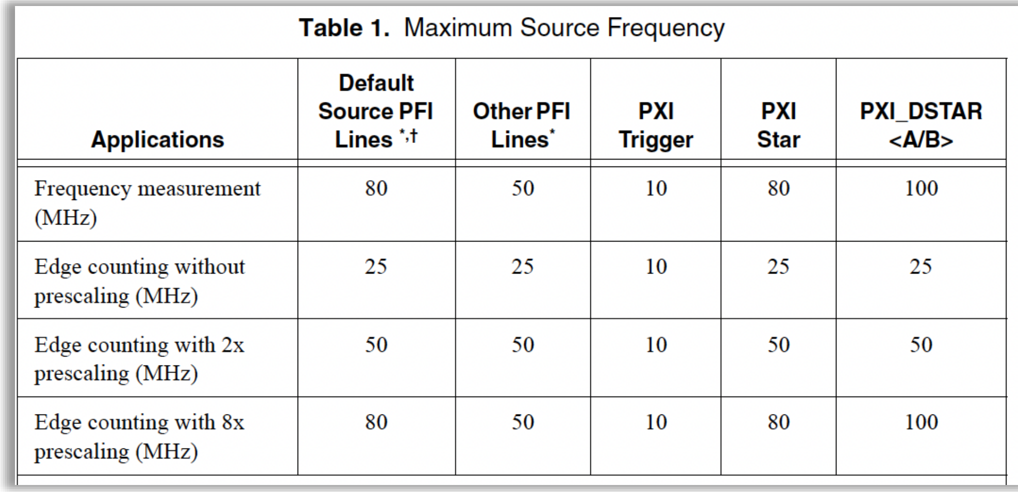


Table 1. Maximum Source Frequency

Applications	Default Source PFI Lines *,†	Other PFI Lines*	PXI Trigger	PXI Star	PXI_DSTAR <A/B>
Frequency measurement (MHz)	80	50	10	80	100
Edge counting without prescaling (MHz)	25	25	10	25	25
Edge counting with 2x prescaling (MHz)	50	50	10	50	50
Edge counting with 8x prescaling (MHz)	80	50	10	80	100

6614 data sheet

This was a turning point

Adding time to the project was a turning point.

Changes made:

- Replaced DataSocket with MQTT (Francois Normandin’s library)
    - Constant broadcasts replaced with messages

But, I inserted it gently.

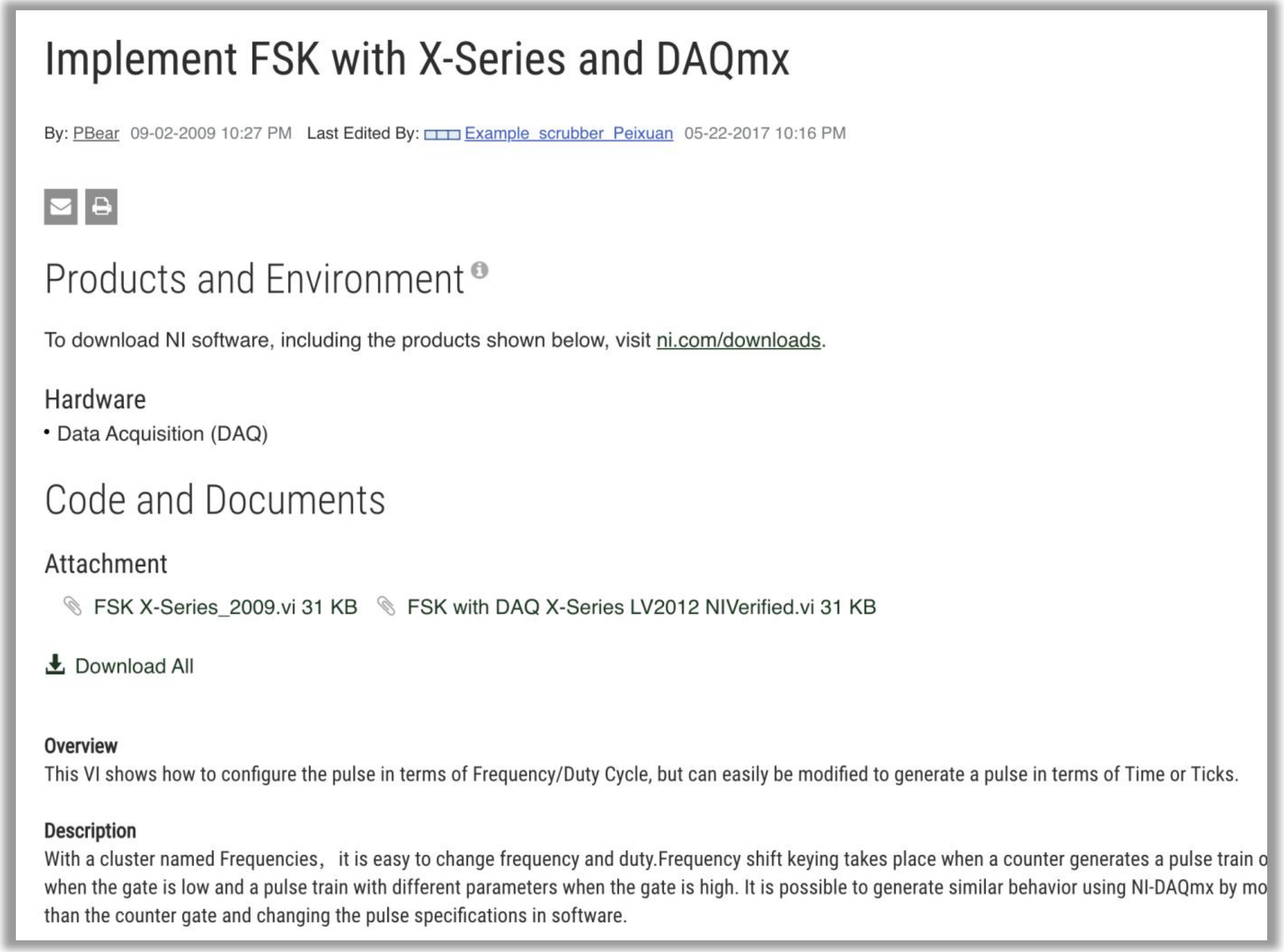
I should have aggressively rewritten both apps in a completely event-driven style.

How that plan failed (3 of 4)

Issues with NI hardware, software, and support

- Translation from Traditional DAQ to DAQmx was mostly easy
- Except when it wasn't
- Very few people at NI know anything about Traditional NI-DAQ
- Took 73 days for NI tech support to figure out a different clock rate issue

Example: Very different approach to FSK in old DAQ vs. DAQmx, only supported on newer hardware



Implement FSK with X-Series and DAQmx

By: PBear 09-02-2009 10:27 PM Last Edited By: Example_scrubber_Peixuan 05-22-2017 10:16 PM

Products and Environment

To download NI software, including the products shown below, visit ni.com/downloads.

Hardware

- Data Acquisition (DAQ)

Code and Documents

Attachment

FSK X-Series_2009.vi 31 KB FSK with DAQ X-Series LV2012 NIVerified.vi 31 KB

Download All

Overview

This VI shows how to configure the pulse in terms of Frequency/Duty Cycle, but can easily be modified to generate a pulse in terms of Time or Ticks.

Description

With a cluster named Frequencies, it is easy to change frequency and duty.Frequency shift keying takes place when a counter generates a pulse train o when the gate is low and a pulse train with different parameters when the gate is high. It is possible to generate similar behavior using NI-DAQmx by mo than the counter gate and changing the pulse specifications in software.

How that plan failed (4 of 4)

Net impact of NI DAQmx translation issues

- Spent months debugging individual counter behavior
- Unable to spend time with big picture testing until the very end
- Lots of time spent investigating workarounds, such as using old hardware, as well as an FPGA solution

Customer was ***very*** understanding and involved with tradeoff decisions.

Regret

Recall my #1 reason to refactor:

Probably lots of undocumented behaviors that need to be retained (e.g., “magic” delays)

Over time, “refactor” became untenable because I retained undocumented behavior

Example: error recovery depended on constant rebroadcasts of DataSocket values

Late in *(ahem, fixed-price contract)* development: introduced more event-driven programming to make operation more intentional

Lessons Learned

- I still believe ”refactor” was a plausible place to start
  - Wrong but plausible
- As things went wrong, I should have been more eager to rearchitect from scratch
  - Example: Over time, I came to understand the system details better than anyone else. Rather than stick with undocumented behavior, I should have thrown it out and rewritten it.
- Bid fixed-price contracts even higher than you think 😉

Huge Thanks

- Kevin Price, NI Counter/Timer Forums “Proven Zealot”
- Sierra Peaks
  - Tim Brooks
  - Justin Whitling
- NI
  - David Bonal, NI Account Manager
  - Leonard Nava, Senior Technical Representative
  - Aaron Long, Field AE