



Certified Experts in Automation Engineering to Design, Control, Test & Adapt

## Beer Bottle Thermal Drinking Simulator Upgrade

### Author(s):

Richard M. Brueggman, Founder & CEO, Data Science Automation, Inc.

Ryan W. Vallieu, Senior Automation Systems Architect, Data Science Automation, Inc.

### NI Product(s) Used:

LabVIEW

LabVIEW RT

LabVIEW FPGA

NI Scan Engine

NI cRIO-9075

NI 9213 (Qty 2)  $\pm 78$  mV, Thermocouple Input, 75 S/s, 16 Channel Module

NI 9403 5 V/TTL, Bidirectional Digital I/O, 32 Channel Module

NI 9870 4 Port, RS232 Serial Interface Module

### Category:

Advanced Research;

Industrial Machinery & Control

### The Challenge

Develop a more robust and capable LabVIEW software platform for characterizing the thermal insulation properties of bottle designs when drinking is simulated.

### The Solution

CompactRIO FPGA hybrid mode was leveraged to provide deterministic data collection along with ease of integration with already existing serial hardware. The CompactRIO processor was utilized to perform time triggered hardware control based on test profiles provided by the operator.

### Introduction

Data Science Automation (DSA) is a premier National Instruments (NI) Alliance Partner that specializes in automating and educating the world leading companies. Clients choose DSA because of DSA's deep knowledge of National Instruments products, disciplined process of developing adaptive project solutions, staff of skilled Certified LabVIEW Architects and Certified Professional Instructors, and unique focus on empowerment through education and co-development.

### Bad Inheritance

The main purpose of the system is to determine the thermal insulation properties of various bottles. The test system in question simulates the warming of the bottle and contents caused when the person drinking from the bottle grasps the bottle and holds it for a certain time period and then drinks. The act of holding the bottle warms the bottle and hence the liquid. Lifting the bottle to drink and tipping it back down stirs the contents. This system aims to simulate that process.

The main hardware for the system was not modified. The hardware consisted of a heater, a fan, a fan tachometer, and a peristaltic pump, thermocouples, and an air pump. Three of the hardware systems were serial based including the pump, the fan, and the tachometer. The heater system was controlled by duty cycle utilizing a closed loop control duty cycle output. The air pump was used to stir the liquid in the bottle was controlled by simple on/off Boolean logic.

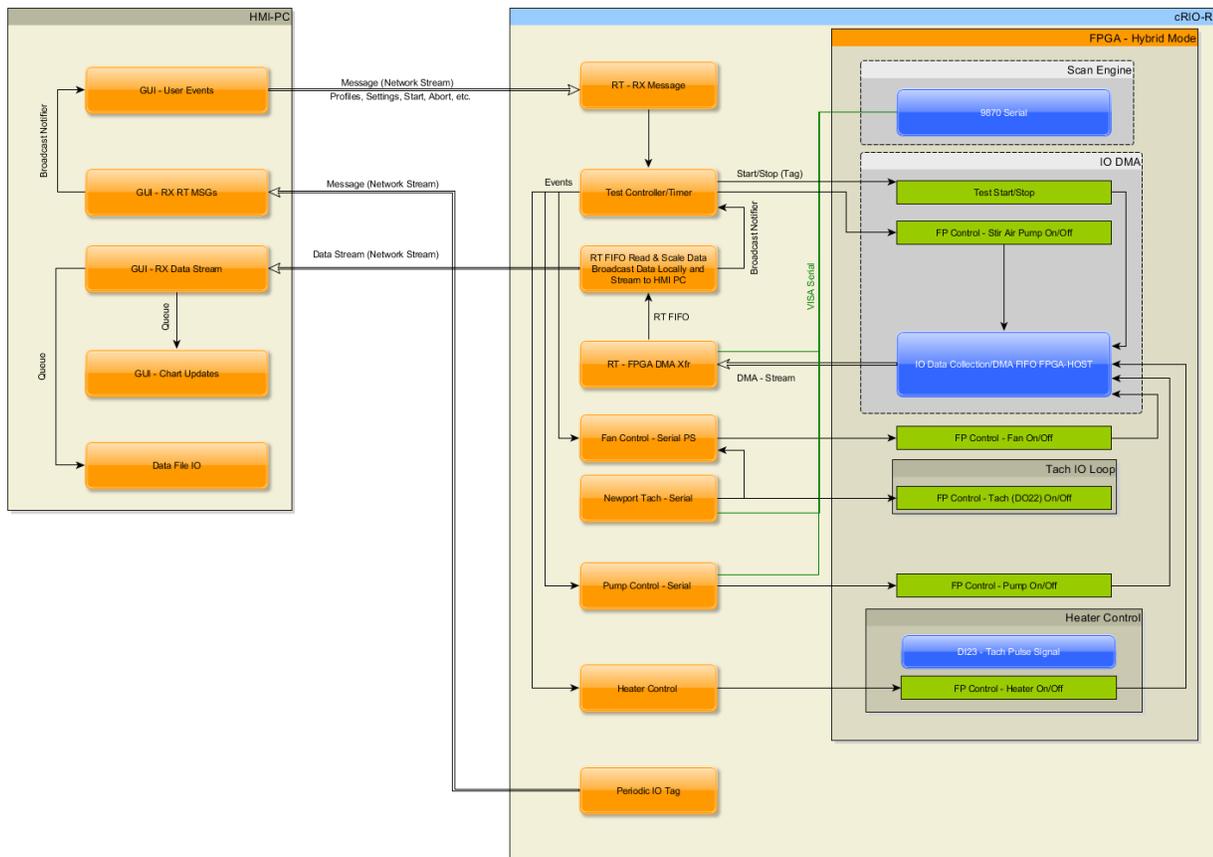
DSA was brought in to rewrite the bottle testing system since it had many issues that our customers wanted addressed. The two biggest complaints with the system were lack of determinism for the data collection periods and wide variations in total profile test time. Both of these complaints stemmed from the poor software architecture selected for the system by the original developer. The other desire of the customer was to have a much higher degree of control over the “drinking” profiles. The original system was locked in to only performing repeat steps that could not differ in event length and periodicity between cycles or events. The heater and fan could not be run during a whole test profile to simulate a person holding a drink and not setting it down.

### **Squandering Embedded Technology**

The original system utilized an NI cRIO-9075 unit as the communication hub for the system utilizing the NI Scan Engine and the NI 9870 serial card and NI 9403 bidirectional digital card. Originally this system was only utilized to provide the serial connectivity to the hardware and control the digital outputs to turn on the various system relays such as the heater power relay. Communications with the devices serial devices were actually performed on the HMI PC and Network Shared Variables (NSV) and the Shared Variable Engine (SVE) were used to communicate between the PC and the cRIO. The only control logic on the cRIO was a loop for controlling the duty cycle of the heater. This design is basically the exact opposite architecture that should be implemented for a system with exactness desired in the sampling rates and profile event timing. The Scan Engine was set to sample the 9213 cards with them set up for the High Resolution sampling, which only updated the reading that were being recorded every tenth of a second with new data every second.

### **Utilizing the True Power of NI Embedded Technology**

A DSA certified LabVIEW Architect discussed the desired system parameters with the client and developed a basic system architecture and communication diagram to demonstrate how the system could be configured to meet the specified requirements as laid out Figure 1.



**Figure 1: New Software Architecture/Communications Diagram**

The FPGA was leveraged to provide precise sampling time control utilizing the 9213 cards in high speed mode. 10 samples were averaged per channel per one-tenth second period. The 9213 is capable of handling sampling rates up to  $1/(\text{Conversion Time} * \text{Number of Channels})$ .

The main data collection loop provided data to the RT host every 100ms. In addition to providing the thermocouple data to the RT host via DMA transfer the Boolean input front panel controls used to control the hardware subsystems was also bundled into the channel data and provided to the host. This Boolean data tracked Test Running, Pump On/Off, Heater On/Off, Stirrer On/Off, and Fan On/Off.

The NI-RIO/FPGA running in Hybrid mode allowed the control and monitoring of the serial instruments to be moved to the 9075. In truth they were always going through the cRIO-9075 even when originally being called from the PC as the NI 9870 was always in the system. Data from these subsystems is not recorded with the main thermocouple data. The ability to leverage the NI VISA for instruments connected to the FPGA definitely reduces implementation time over having to create the protocols on the FPGA itself. Driver software written for the PC could easily be modified for running on the RT target with minimal changes.

This architecture provided a stable event-timing based system. The test control and module was made much more deterministic in this system versus the original. In addition to controlling all of the instrumentation, the original system was only written using a single state machine that was controlling all subsystems as well as monitoring for front panel button interactions and logging data. The loop PC main application loop contained multiple User Event structures that would interrupt the test sequence timing and data acquisition timing.

### Lossless Logging

NI Network Streams were chosen to send the buffered data to the HMI program for display and logging. In the original system all data was plotted to waveform graphs and only after the test was complete would the system record the data to file. Building the data by appending the waveforms and then updating the graphs on every acquisition iteration also severely impacted the system performance. The graph had to be redrawn and the system memory manager was invoked when appending multiple channels of data into multiple waveform arrays that were not pre-allocated. The updating of this chart was also occurring during all of the other system control, communications, front panel interaction monitoring, etc.

The new system leveraged buffering on several levels to ensure lossless data logging. The Network Streams from the RT system to the HMI were buffered. Queues were utilized to leverage First-In-First-Out buffering to the logging module. This allowed data to be buffered and then logged when a block of data was collected – providing more efficient writing to the hard drive. Writing the log file while the test was running also protected the system from data loss. Waveform Graphs were replaced with Waveform Charts preconfigured for the amount of memory needed to display the longest test cycles.

### Event Control

One of the main desires of the operators of the system was to be able to specify arbitrary timed event steps instead of being limited to repeating exact cycle steps evenly over time. It was desired to be able to run the heater and fan continuously throughout a run, and also control timing of each piece of hardware independently based on time within a complete test.

DSA was able to provide a new test profile interface that allow individual control of the hardware based on time and on/off events. This new method of describing the test procedure would still allow for replicating the original system testing profile of repeating test cycles, but also allowed for arbitrary cycle length and on/off control. Visualization was also provided to show the created cycles for review by the operator.

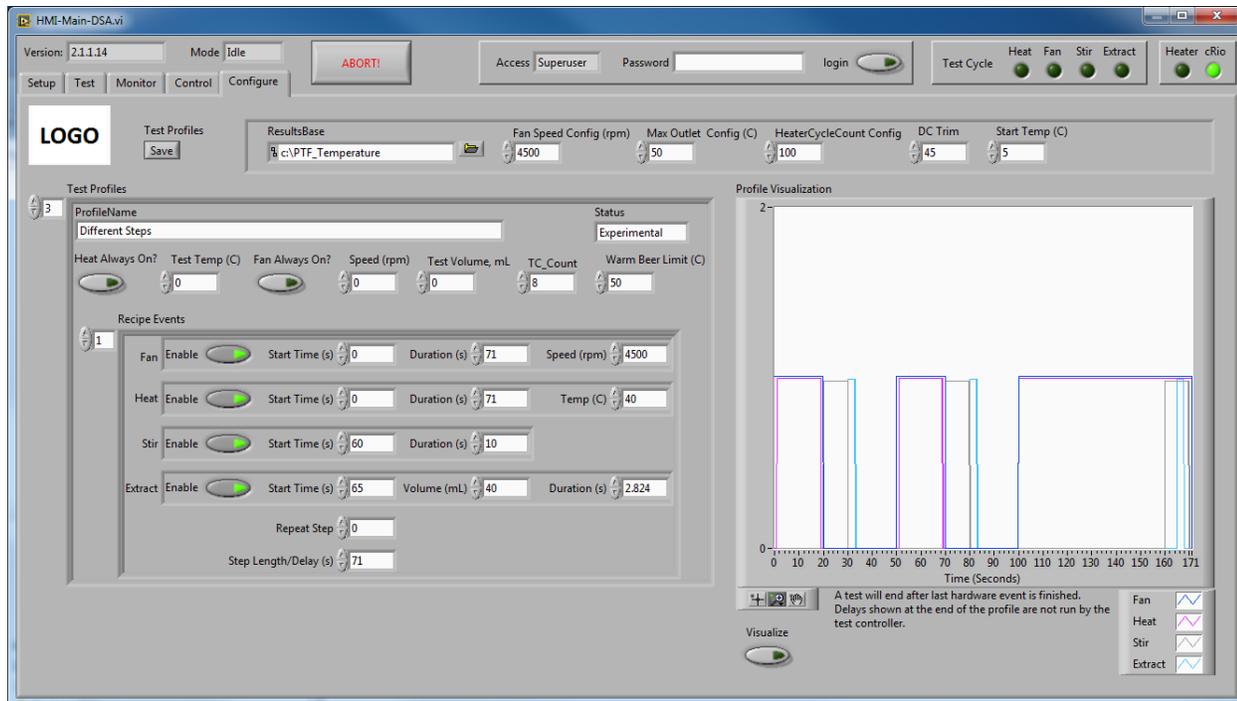


Figure 2: New Profile Generation and Visualization

Figure 2 shows an example profile that can be created utilizing the new profile interface. The operator has the ability to specify Recipe Events using an array of independent steps, delays, and repeating any steps as prescribed. In this example a

repeat of the first index (index zero) prescribed is shown with the two repeating cycles at the beginning of the test, followed by a cycle with a single step description of turning on the fan and heat for the entire cycle length and only extracting the fluid at the end of the step. None of this level of control was possible before in the original architecture. LabVIEW made this easy to visualize and convert to a series of timing events with On/Off Times extracted and the matrix of On/Off times sent to the cRIO Test Controller module.

### **Adaptability and Maintainability**

Due to following programming standards that DSA teaches in LabVIEW classes and recommended best practices – this system can now grow or be modified with minimal recoding of the system. The HMI architecture is agnostic to the type of instrument that is connected to the cRIO for control of the pump, the fan, and the tachometer. In fact the HMI code is not tied to any one particular model of cRIO in any way, meaning the customer can upgrade the embedded system for more powerful devices in the future while only recompiling the FPGA for the new target and updating the FPGA reference in the RT code.

The real-time test controller code also does not directly control the instruments on its block diagram. Modular code was written to minimize the amount of recoding on the RT system and if more deterministic systems that utilize analog control are desired only the specific instrument code which is highly cohesive needs to be modified. Messaging architectures were leveraged to allow the instruments to be coded in a modular fashion.

### **Conclusion**

The combination of the FPGA and deterministic Real-Time OS of the cRIO made the desired upgrade goals of tighter timing constraints possible. Separating the HMI code from the embedded code and utilizing buffered acquisition techniques guaranteed lossless data collection and means the HMI updates and operator interactions will not affect sample timing.

### **Contact Information**

Richard M. Brueggman

Ryan W. Vallieu [info@DSAutomation.com](mailto:info@DSAutomation.com)

Data Science Automation Inc., 375 Valley Brook Road, Suite 106, McMurray, PA 15317-3370

[www.DSAutomation.com](http://www.DSAutomation.com)