Reconfigurable Pulsed Laser Deposition Control and Monitor System

Authors:
Richard Brueggman, President & CEO, Data Science Automation, McMurray PA
Quintin Stotts, Senior Automation Systems Consultant, Data Science Automation, McMurray PA

NI Products Used:
LabVIEW 2012
USB-6220

Category:
Advanced Control Systems

The Challenge

Provide a pulsed laser deposition (PLD) control application that will differentiate a company from the competition by offering an aesthetic, full-featured software package to monitor and control their PLD systems.

The Solution

Develop a modular LabVIEW solution that allowed for a single application that could be used with various PLD configurations by dynamically enabling/disabling certain modules for each deployment.

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.

A leading Pulsed Laser Deposition company selected DSA to develop a LabVIEW control and monitoring application because they felt the software would get developed much faster and be much more flexible and adaptable versus taking on the project internally.

Multiple Hardware Interfaces and Technologies

LabVIEW was selected as the best programming language for this project due to the requirements to have a user friendly graphical user interface and the need to interface with various hardware components, both NI and non-NI. The software was required to manage various aspects of the Pulsed Laser Deposition systems:

- Laser Scanning. By controlling two motors for positioning, the high power laser of the PLD system could be directed at any point on a semiconductor target in the PLD chamber. LabVIEW controlled the motors through ActiveX controls. Various scanning profiles could be programmed via the LabVIEW GUI, as well as manual positioning of the laser thru an easy to use pop up jog screen.
- Substrate Rotation. The substrate, which receives the deposit from the vaporized target, is rotated into position by controlling a geared third motor. LabVIEW controlled this motor via a RS-232 interface.
- Temperature Control. A Eurotherm hardware controller controlled the chamber temperature, though this control aspect could have been handled entirely in software using the LabVIEW PID Toolkit. LabVIEW communicated to the Eurotherm easily via a RS-232 interface using a LabVIEW Instrument Driver found in the NI Instrument Driver Network.
- Mass Flow Controllers. LabVIEW easily interfaced with this hardware using the USB-6220 multi-function DAQ device. Digital outputs enabled/disabled the MFCs, while analog outputs set the flow set points.

- Target Selection. The system could have 4-6 different targets that the laser could strike situated on a turn table. Selecting which target was to be positioned in front of the laser beam was controlled with a fourth motor. This was again interfaced with LabVIEW via RS-232.
- Laser Pulse Control. The high power laser was able to accept a TTL pulse train to control how many pulses the laser output and the duration of each pulse. The counters on the USB-6220 again easily handled this task.

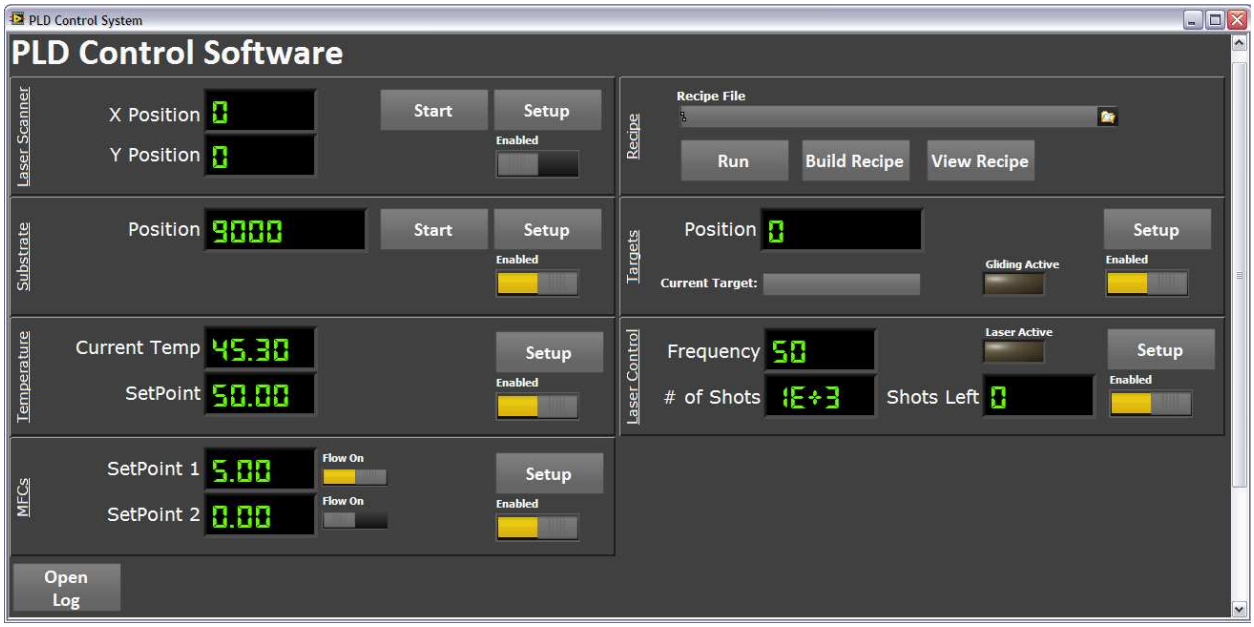

Figure 1 - Software with all modules enabled.

Using the NI USB-6220 as the main hardware interface for several of the modules simplified the code development versus developing LabVIEW code for a non-NI hardware DAQ device. The USB-6220 contained enough DAQ components and channels to control the Mass Flow Controllers and Laser pulse control, with spare channels remaining to handle future expansion of the system.

The main LabVIEW GUI contained pertinent status indicators for each module as well as setup pop up screens for each module to manually control the module and/or configure settings. However, to bring everything together, a Recipe module was implemented to allow automated control of the PLD system. It was mainly used for stepping through the various targets and pulsing them with the laser, but could be expanded to incorporate control of all of the available modules.

A GUI Rooted in Adaptability

This LabVIEW application had to be very modular since each new deposition configuration could different. For example, one system may need the software to only control the scanning of the laser and temperature control while another configuration may require the software to control scanning of the laser, substrate rotation and control of the laser pulses. The PLD company did not want to have to worry about modifying the code every time a new system was constructed, even though they had in house resources capable of modifying the code due to prior LabVIEW training. To handle this challenging requirement, a flexible LabVIEW architecture was set up that would determine what modules to enable/disable upon start up. This way, the same application would always be installed for each customer, but the PLD company would just provide a unique file for each customer that would only enable the modules requested (and paid for).

VI Server, Simplicity.
To provide a very simple way to enable/disable modules, the power of VI Server in LabVIEW was leveraged. The main GUI of the PLD Control System was architected so that each module's group of controls and indicators resided on a dedicated tab control (each tab control only had a single tab). When a given module was disabled, the code would simply set the visible property of the tab to false, making it, along with all the module controls and indicators, invisible on the GUI. When a module was enabled, its tab would be visible (see Figure 2). The unique file that each customer received contained information to enable/disable the modules, without that file, the software would not be able to run. Taking it a step further, code was put in place to reposition the tab controls and set the bounds of the GUI to optimize the screen layout after enabling and disabling the configured modules. This allowed for a very professional, clean GUI without any strange blank spaces of the GUI.

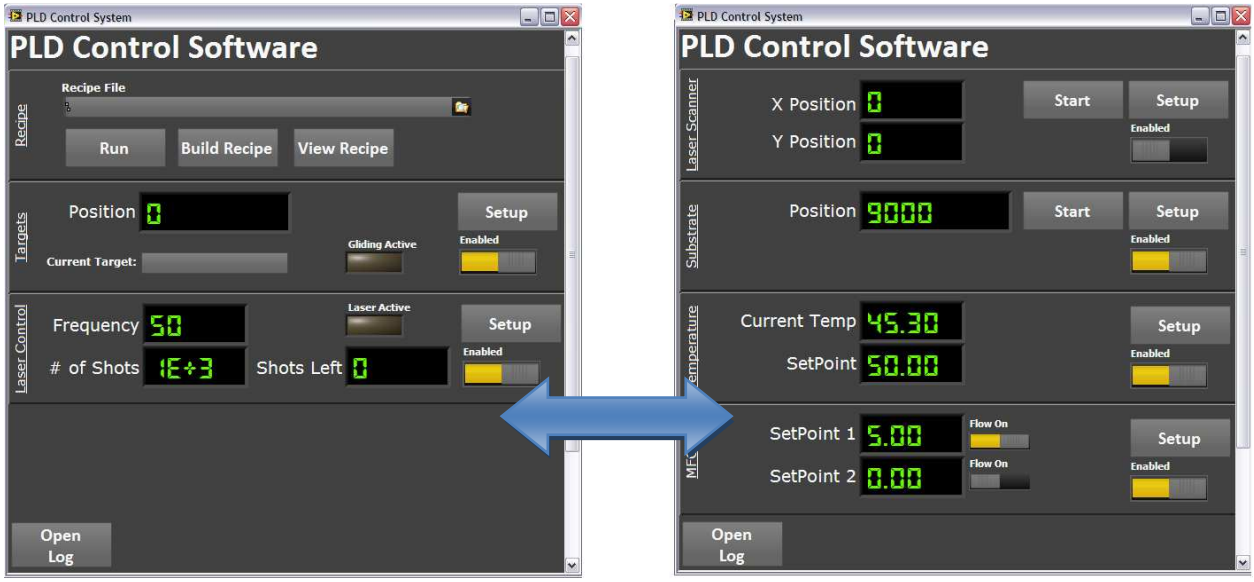

Figure 2 - Same software, two different configurations.

Conclusion
LabVIEW allowed for the development of an extremely modular, low maintenance application, while expediting the development of the system. The entire application was completed and tested with less than 4 man-weeks of effort. LabVIEW was able to easily interface with different devices using different technologies: ActiveX, RS-232 and NI-DAQmx.

Contact Information
Richard Brueggman, rmb@DSAutomation.com
Quintin Stotts, qrs@DSAutomation.com