

## **Lab Automation Software**

Alan H. Katz, PhD  
Chief Scientific Officer  
Hudson Control Group

### **Introduction**

Recent software advancements have made it much easier to automate almost any procedure typically carried out in a drug discovery laboratory. It is now possible to design “virtual workcells” from a small, dedicated collection of liquid and microplate handling robots to carry out the simplest or most complicated procedures. This tutorial will show how this approach makes it possible to build a flexible system for testing drug candidates with a wide variety of protocols. Similarly, various molecular biology procedures often used in the manipulation of proteins and nucleic acids can be run in a single workcell. Now that the setup of new protocols is so easy, automation can be applied to many tedious processes that previously weren’t considered possible.

### **Instrument Setup**

First, one has to setup the equipment from which the scientist can choose. To maximize flexibility, one should follow two principles: 1) include all of the equipment that one might need, and 2) make sure a microplate can be readily transported between any two pieces of equipment. In designing our own systems, we also like to make sure each instrument is easily accessible to the user, so that it may be used in one-off, manual experiments.

The key components that allow plate movement among instruments are robotic arms (in the following examples, a PlateCrane from Hudson Control Group), and moving tracks (such as Hudson’s LabLinx). Instruments that are frequently used in sequence benefit from contiguous placement along a track. Other components, and those that can’t be adapted to tracks, must be within reach of the robotic arm.

### Example One: Biological Assay Screening System

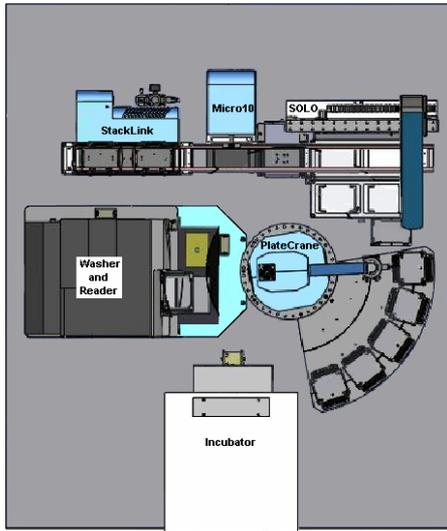


Figure 1. VaryScreen system to support screening protocols

The system shown left was designed to carryout a large number of screens normally encountered in drug discovery research. A multimode reader (e.g. Biotek's Synergy 4, or BMG LabTech's PheraStar) supports the vast number of protocols involving Fluorescence Intensity, Polarization, and resonance energy transfer (FRET), as well as AlphaScreen, ELISA, luminescence and UV/VIS absorption-based assays.

The system includes a multi-channel dispenser (MicroFlex), an automatic pipettor (SOLO), and a searchable stacking system connected by a LabLinX track. In addition to the multimode reader, the system also contains an automatic incubator (Liconic STX40) a microplate shaker nest, and a microplate washer (ELx405) accessible to the PlateCrane robot arm.

### Example Two: Proteomics and Genomics WorkCell

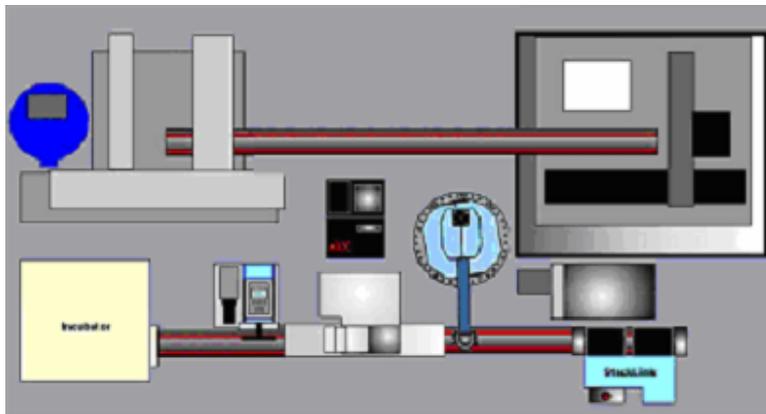


Figure 2. A ProLink Proteomic and Genomic workcell.

A larger system, shown below, can be used to carry out a wide range of molecular biology functions, such as protein expression and DNA purification. In addition to the components of the previous example, it contains a colony picker, plate sealer, centrifuge, thermocycler, and an automatic vacuum filtration system.

### Software Integration and Virtual Workcell Design

At Hudson Control Group, we have developed workcell scheduler software called SoftLinX For Systems, which is also a tool for building virtual workcells from an integrated workcell, such as the systems described above. After the equipment is arranged in the lab, the corresponding software interfaces are loaded into SoftLinX.

Some of the equipment will require individual configuration, such as pinning down the location of microplate nests, or the type of plates and pipette tips being used. After this is complete, SoftLinx is sufficiently aware of the system to make it ready for protocol design.

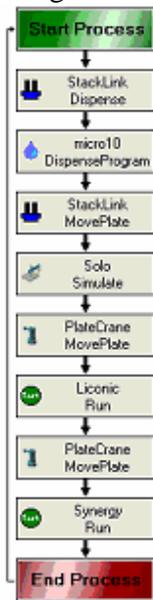


Figure 3. A simple automation protocol

The user will then open the method editor and design virtual workcells simply by building process flowcharts. The editor includes icons for each instrument, which are dragged into the growing process, and appropriate parameters are defined (such as dispense volume or incubation temperature).

In the simplest cases, the entire protocol can be included in a single process, such as the BCA UV protein assay shown at left. More commonly, a protocol consists of a series of parallel processes, each focusing on a particular portion of the method.

Each process can be set to start as soon as the protocol is run; or it can be set to wait until any number of conditions are met, such as the presence of a plate in a particular instrument nest, the identification of an active compound (see example One, below), after a number of identified hits is reached (example Two), or after another process is complete.

### Processing Logic and Smarter Automation

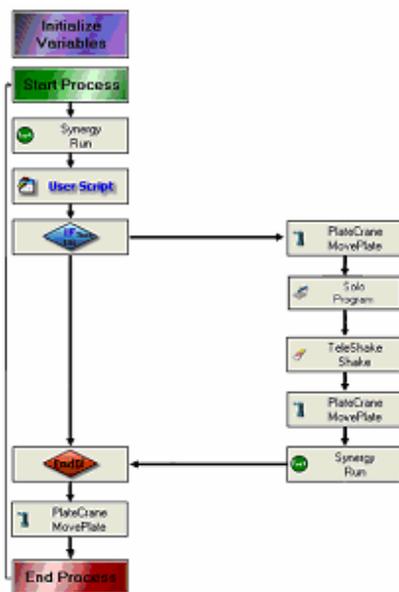
Many frequently used protocols can be automated with the set of tools just described. However, the software also adds the ability to introduce programming logic into a process, greatly increasing the power and flexibility of automated protocols. For example, at the start of a process, one can create and set global variables so that every process can be aware of the current state of the entire system. A Boolean variable can remain false until a condition is met, or a desired result is obtained. A numeric variable can be used to keep track of how many times a particular plate has been washed, or the number active compounds have been identified (example Two). To take full advantage of these variables, an If..Then..Else conditional tool is included to allow one to control how a protocol proceeds based on various conditions. Another basic programming construct, the “While” statement, is also employed within SoftLinx to allow portions of a process to continue (or stall) as long as a logical statement is true.

But if one really wants to add ultimate flexibility as well as unprecedented levels of intelligence to a protocol, there is also the ability to include User Scripts. With full access to Microsoft’s Visual Basic for Applications (VBA), one can write very detailed programs that can carry out all sorts of numerical, string, or even inter-application communication. These scripts allow an automated process to analyze the results being

obtained on-the-fly, and react accordingly. A few examples are included below to demonstrate the sort of functionality that can be added to a protocol.

### Example One: Re-Reading a Microplate Only When Hits are found

There are many times in which it is desirable to carry out additional steps on samples that show interesting results in an initial screen. These steps can vary in complexity from a simple re-reading of the plate, to submission through an entirely different assay protocol. Regardless of the nature of this follow-up test, we first need to establish the presence of a hit in a plate. This is easily done by combining a user script with an If..Then..Else control statement.



In the simple example shown on the left, a spectroscopic study has been carried out on a BioTek Synergy 4 plate reader. After a run, the Synergy 4 creates a text file containing the results from a procedure defined with BioTek's GEN5 software.

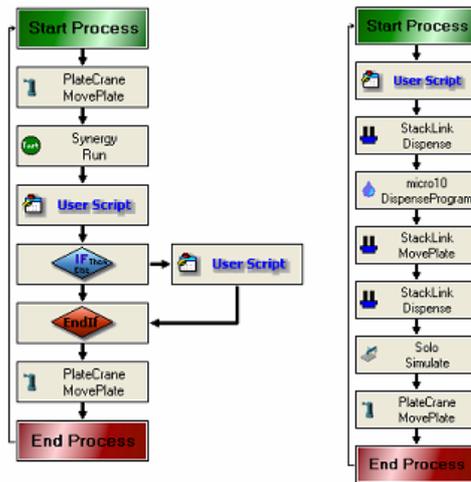
A User-Script is then created to import the contents of the data file and store the results into a matrix of variables representing each well. A Boolean variable, which is initially set to false, is then set to true when a result is found in a well with the appropriate level of activity; the PlateCrane moves the plate to the deck of the SOLO pipettor when hits are identified. The SOLO adds a reagent, the mixture is shaken, and the plate is then returned to the Synergy 4 for the new test. If hit is not found, the variable remains false and the PlateCrane is called to remove the plate.

Figure 4. User scripts and conditional logic in an automation protocol.

Of course, the complexity of the sample manipulation, as well as the nature of the reader that is used to generate the second generation of data can be modified to any extent available to the researcher. This can be of great value in situations in which the second-generation experiment is very slow and can't be used in screening large compound libraries. Instead, this type of procedure allows one to limit the slower assay to samples that merit further investigation based on a primary reading.

## Example Two: Interactive Cherry-Picking

The above examples can lead to increased efficiency, but only if a limited percentage of hits are found. If the density of hits is high enough, it is likely that one hit will be found in every plate, and the above procedure would then require *every* plate to undergo the second-generation manipulations described, which may prove cumbersome.



One way to improve upon this is to build in an automated cherry-picking step to the protocol. A simple example is given here.

In this case, the user script inside the If...Then...Else statement simply keeps track of the exact location, and number of each hit. A second process is added which is set start when hit count reaches the number needed to fill a new plate. The process then searches the stacks for the plates containing the hits, and systematically builds a new plate, along with the appropriate amount serial dilution for computing dose-response curves.

Figure 5. Interactive Cherry-Picking

## Example Three: Assay Design

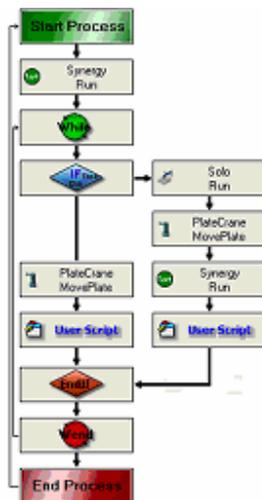


Figure 6. Assay design protocol

This last example demonstrates how a researcher might combine the type of If..Then..Else logic described above, with a While statement to systematically modify the conditions of an assay step until the best results have been achieved.

In this example, we are interested in the effect of solvent conditions on the degree of hybridoma binding between antibody and an antigen. The process begins with a Boolean variable set to false. We then hit the beginning of a While...While-End loop set to run until the Boolean is true. A user script is then employed as above to extract and analyze the results from the fluorescence reader. The conditional statement that follows checks the fluorescence intensity, but this time, an unsatisfactory result leads to the addition of additional reagent and the sample is submitted again to the multimode reader. This process continues until a satisfactory result is obtained and the Boolean is changed to True and the While-End step can be passed. One can optimize a variety of assay conditions by introducing a similar loop for each condition of interest.

## **Conclusion**

There are many additional ways in which software can be used to automate complex multi-step laboratory protocols. Similar approaches can be applied to turn the molecular biology workcell shown above into a series of virtual workcells capable of preparing and producing proteins and nucleic acids. The two instrument groups can even be combined to form a complete assay development system that can support all stages of preclinical research. Hopefully this discussion will spark ideas into ways in which lab automation software can help the reader get the most out of their particular group of instrumentation.