Evaluation of Lookout™ versus iFIX™ for use in Irrigation Modernization Projects

Kurt Colvin
Industrial and Manufacturing Engineering
Cal Poly, San Luis Obispo, CA

September 1, 2001

Abstract:

A systematic evaluation was performed on two Human Machine Interface software packages (Lookout from National Instruments and iFIX from Intellution). This was accomplished with a simulated, automated water level control system, using a Control Microsystems SCADAPack PLC. Through the evaluation, Lookout, clearly outperformed iFIX in nearly all categories. The results of this project were presented to the Cal Poly **Irrigation Training and Research Center** to assist its irrigation modernization program and potentially, as added content in its irrigation management course offerings.

Introduction:

The Irrigation Training and Research Center (ITRC) at Cal Poly is tasked with analyzing the efficiencies of irrigation districts, providing consulting services for the modernization of these districts and offering training courses on a wide range of irrigation management topics.

One of the challenging aspects of irrigation system modernization is the development of effective, consistent and comprehensive automation system user interfaces. An example of this type of an interface would be the computer terminal where an irrigation district operator monitors and controls the automated systems in the field, such as current water levels, pumps and valves. More specifically, the system is known as “SCADA” for Supervisory Control And Data Acquisition and the software used to develop the user interfaces is known as “HMI” for Human Machine Interface. Unfortunately for irrigation districts, there are tens of competing companies that offer a wide range of SCADA and HMI products from completely integrated, turnkey systems, to a single system component. This makes the task of choosing the right products for a specific application very difficult for most resource-constrained irrigation districts.

Therefore, as an attempt to assist irrigation districts and simplify their system modernization decisions, this project was initiated to evaluate the two HMI software packages. It is anticipated that this project, and the projects that will hopefully succeed it will help ITRC and its customers make better decisions when choosing commercial products for irrigation system modernization. Additionally, it is anticipated that some of this material will be integrated into the short course content offered by the ITRC.

Objective

The objective of this project was to evaluate two commercial HMI software packages for use in an irrigation modernization role. The two packages selected were iFIX (by Intellution) and Lookout (by National Instruments). The criteria for evaluation was:

1. Installation
2. Licensing
3. Loading I/O Drivers
4. Tag creation
5. Creating displayable objects
6. Overall development functionality
7. Help files, support and tutorials
8. Overall ease of use

Each package was used to build identical user interfaces for an experimental automated system. Each interface was equivalent in functionality and very similar in appearance. Through this process, experience with the intricacies of each package was used to form an overall opinion of product with the final goal of giving irrigation districts feedback on each of the HMI software packages.

Method:

Automated System Setup:

This project was performed by first configuring a simulated, automated water level control system (see Figure 1). While this may appear to be an over-simplified system, it does represent the basic components of all automated irrigation systems including: a PLC, analog input and discrete input and output. While, of course, actual irrigation control systems are much more complex, they essentially contain these fundamental components. By utilizing such a straightforward hardware setup, it removed many potential variables from the system and allowed for concentration on the operation and functionality of the HMI software.

The hardware environment consisted of the following components (See Figure 2.):

- 1 Controller: Control Microsystems SCADAPack PLC
- 1 Power Supply: Hammond Manufacturing power supply
- 1 Analog Input: Linear potentiometer
- 2 Discrete Inputs: Standard momentary pushbuttons
- 1 DB9-DB9 serial cable
- 1 Dell OptiPlex PC (400 MHz, 128 MB of RAM, Windows NT 4.0)

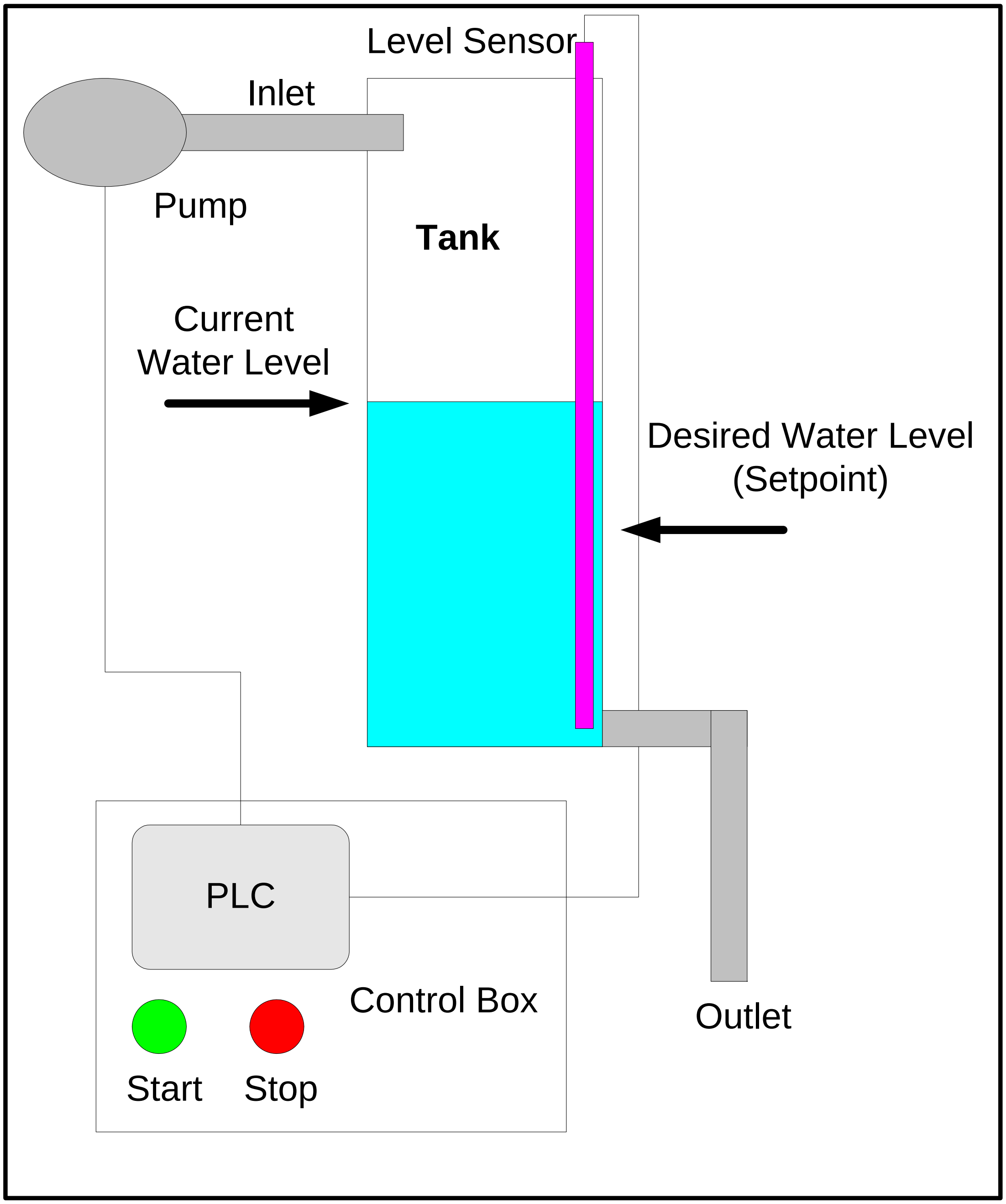


Figure 1. Conceptual Representation of an Automated Water Level Control System

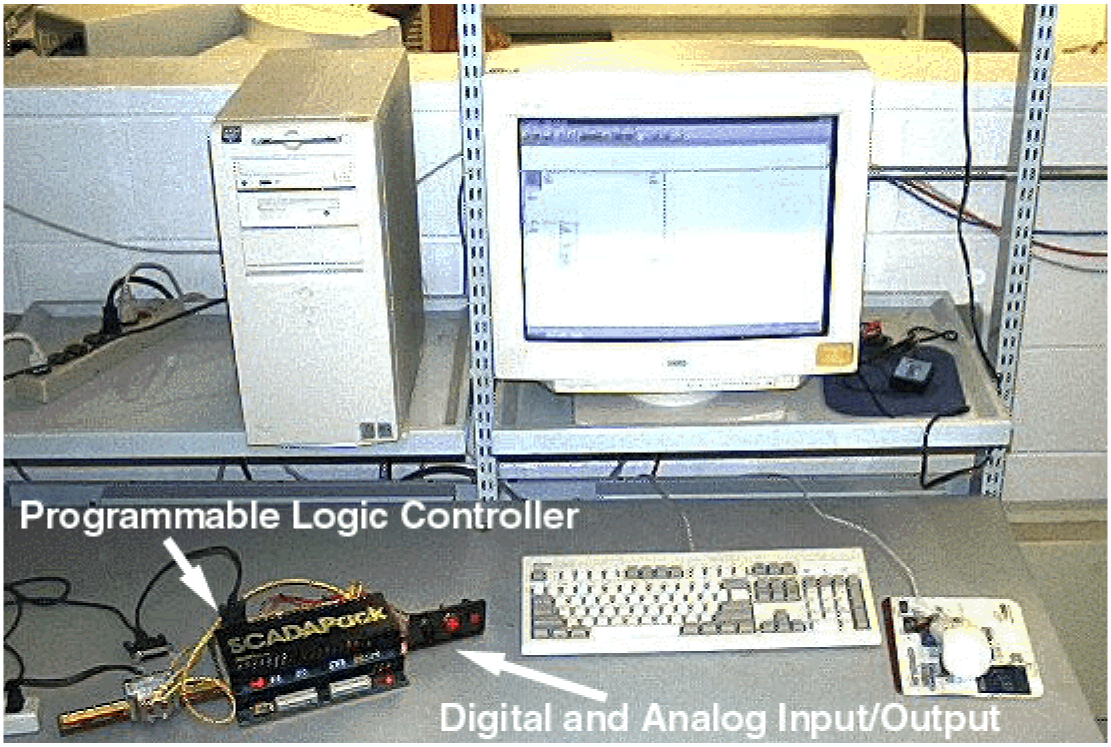

Programmable Logic Controller
SCADAPack
Digital and Analog Input/Output

Figure 2. Automated System Hardware Setup

In addition to the basic hardware setup, a control program was written to simulate the logic that automatically controls the simulated water level. This was accomplished with a ladder logic programming software. Ladder logic is a standard programming software used to program a PLC and it is almost always available from the manufacturer of the PLC. See Appendix A for documentation on the ladder logic control program.

It is important to understand that this system is fully automated at this stage. It is designed to maintain the tank water level at the current setpoint without any human interaction. However, if any of the system's parameters need to be modified, such as the tank level setpoint, it is necessary for the system to have functionality to accomplish such a change. Additionally, as with almost all automated systems, one of the human's roles is to monitor the state of the system, and take appropriate action if it is required (such as system shutdown in an emergency). These two functions, monitor and control, are the two primary roles of the automated system's user interface.

Software Environment:

Once the automated system was configured, programmed and operational, the user interface was developed using the two selected HMI packages. From the user's perspective, the operation such an automated system interface should be very straightforward: The PLC maintains the water tank level at a specific setpoint. The setpoint is merely a variable maintained in the PLC that should be able to be modified at any time by the operator. Additionally, the status of other system parameters, such as: current water level, pump status, system status and push button state should all be present in the interface. Finally, a method of issuing an alarm if select system parameters reach threshold values is a standard HMI component and should be included. These were the specifications used to create the user interface using the HMI packages.

Building the Interface using Lookout version 4.5

1. Introduction

Lookout (Figure 3.) is a HMI packaged made by National Instruments (NI). NI is a large hardware and software company that is probably best known for its primary measurement product called LabVIEW. LabVIEW used to collect, process and record data from a wide variety of scientific instruments and uses a visual programming language called G. Lookout is NI's HMI offering for the SCADA/automation industry.

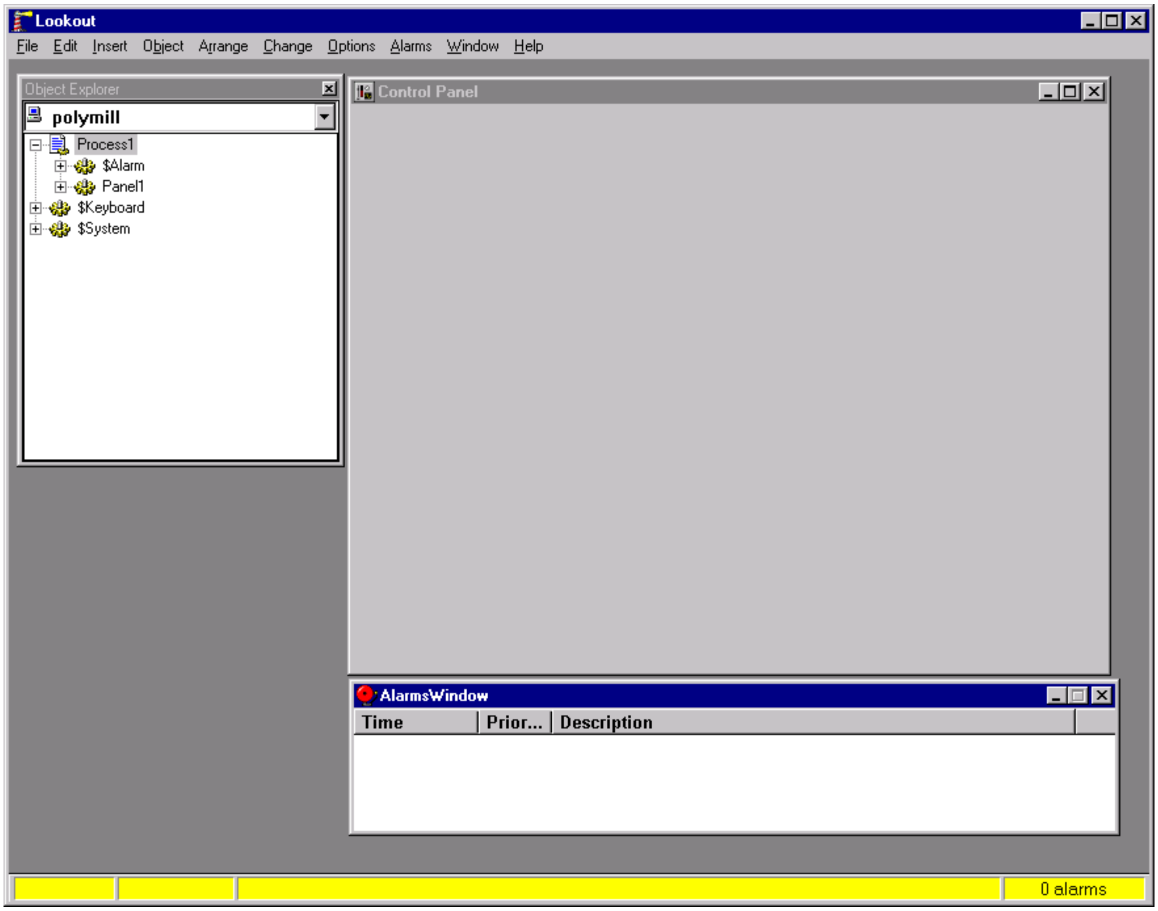


Figure 3. Lookout 4.5 Development Environment

2. Installation

Installation of Lookout was very straightforward. After the Lookout CD was inserted, the Autoplay started the installation program and led me through an installation wizard where I answered a few questions and inserted the license serial number. (If Autoplay is not enabled, start the **lookout.exe** program on the CD) Lookout does not require a hardware key, but does all its licensing within the software. The default directory where Lookout is installed is: C:\Program Files\National Instruments\Lookout.

3. Loading I/O Drivers

Loading the I/O drivers allows for the HMI software to communicate with the PLC. This is often a critical point in the selection of a HMI package, as an appropriate driver for a specific PLC is simply not available. Luckily, for this project, the Control Microsystems SCADAPack used a very standard addressing scheme known as Modbus, and Lookout included a serial Modbus driver.

In actuality, it was required to first build a display panel in Lookout before loading a driver. To do so, jump a head to Building the Main Panel. Once the panel is created, return here and follow this procedure to install the Modbus driver:

1. Open the Object Explorer: Select Object Menu -> Object Explorer
2. Right click on the Main panel and select New Object
3. Open the Drivers folder and select the Modbus driver
4. Fill out the next dialog box, selecting a name for the device (i.e. SCADAPack) and the appropriate serial port and select OK.
5. The PLC driver has now been loaded and you can access addresses within the PLC.

4. Building the Main Panel

Lookout uses an Object Explorer to display many different object types (processes, panels, text boxes, buttons, etc.). When Lookout starts, it already has two objects configured ($Keyboard and $System). These objects will always be visible in the Object Explorer should not be altered. First, a Process must be created. A Process is simply a collection of objects that will eventually make up the interface. To create a Process and the control panel:

1. Select the File Menu -> New
2. Enter the Process name: “Main” and select OK.
3. Enter OK to take the default display properties of the new control panel.
4. You have now created a process “Main” which includes a default control panel: Panel1.
5. You can right click on the panel to rename it at any time.
6. Now that you have created a panel, you can return to Loading I/O drivers at this time to load the Modbus I/O driver.

The next step was to create “tags” or as Lookout calls them, “Aliases.” Tags are merely common names to represent memory locations on the PLC. For example, the #0 input address on the PLC is at address 10001. In our example system, this input was connected to the Stop button. Therefore, a tag was created called STOP, and that tag was mapped to the address 10001 on the PLC. This is a very standard practice using HMI software. To create the tags:

1. Right click on the I/O driver object (SCADAPack), and select Edit Database.
2. Enter the following Alias names and corresponding PLC addresses. Click save after each entry:

Name	Address
STOP	10001
START	10002
PUMP	00001
TANK_LEVEL	30001
SETPOINT	40001
SYSTEM_STATUS	00002

3. Select the Quit button when all tags have been entered.

At this point, there is a main panel for the user interface, the tags have been created and the communications driver has been loaded, but there are no graphical objects to represent the system. The following instructions walk through each of the graphical object creation method.

Discrete Input Display:

1. Right click on the Main panel and select Insert Expression.
2. Navigate through the object tree and locate the STOP tag.
3. Double click the tag to insert it into the yellow field at the top of the insert expression dialog box.
4. Hit OK, and select the default display properties.

The graphical button now represents the position of the physical stop button wired to the PLC. When the button is physically pressed, the graphic changes color. The colors, appearance and other properties of the graphic can be customized by right clicking on the graphic and selecting Display Properties.

Build the other discrete inputs of the system in a similar manner: START, SYSTEM_STATUS, and PUMP

Analog Input Display as a Digital Number:

1. Right click on the Main panel and select Insert Expression
2. Similar to the previous step, Navigate through the object tree and locate the TANK_LEVEL, and click OK.
3. In the next dialog box, select OK, to take the default display values.

This object now displays the current value that the PLC is obtaining from the potentiometer (simulated water level in the tank). Again, the custom display properties can be modified if desired.

Analog Input Display as a Linear Bar Display:

1. Right click on the Main panel and select Insert Expression
2. Similar to the previous step, Navigate through the object tree and locate the TANK_LEVEL, and click OK.
3. In the next dialog box, select display style as “bar up.”
4. Next, input the minimum units: 7000 and the maximum: 32000, then OK. There are several ways to scale these units, but we will use the raw numbers at this time.

5. The display now represents a tank with a water level that varies according to the analog input (turn the potentiometer to see the level change).

Data Entry Box:

1. To insert a data entry box, right click on the Main process in the Object Explorer.
2. Select New Object, and navigate to the Control folder.
3. Locate the TextEntry object and select OK.
4. In the New Text Entry Box, enter a name, prompt string, and select the "Remote" text source.
5. Right click in the URL field, and locate the SETPOINT tag, and select OK and again, OK.
6. You have now created a new Text Entry object, but not yet a graphical display.
7. Click on the text entry object in the object explorer, and drag it to the main display. Select any custom properties and hit ok.
8. Using the handles on the text entry object, size it appropriately.

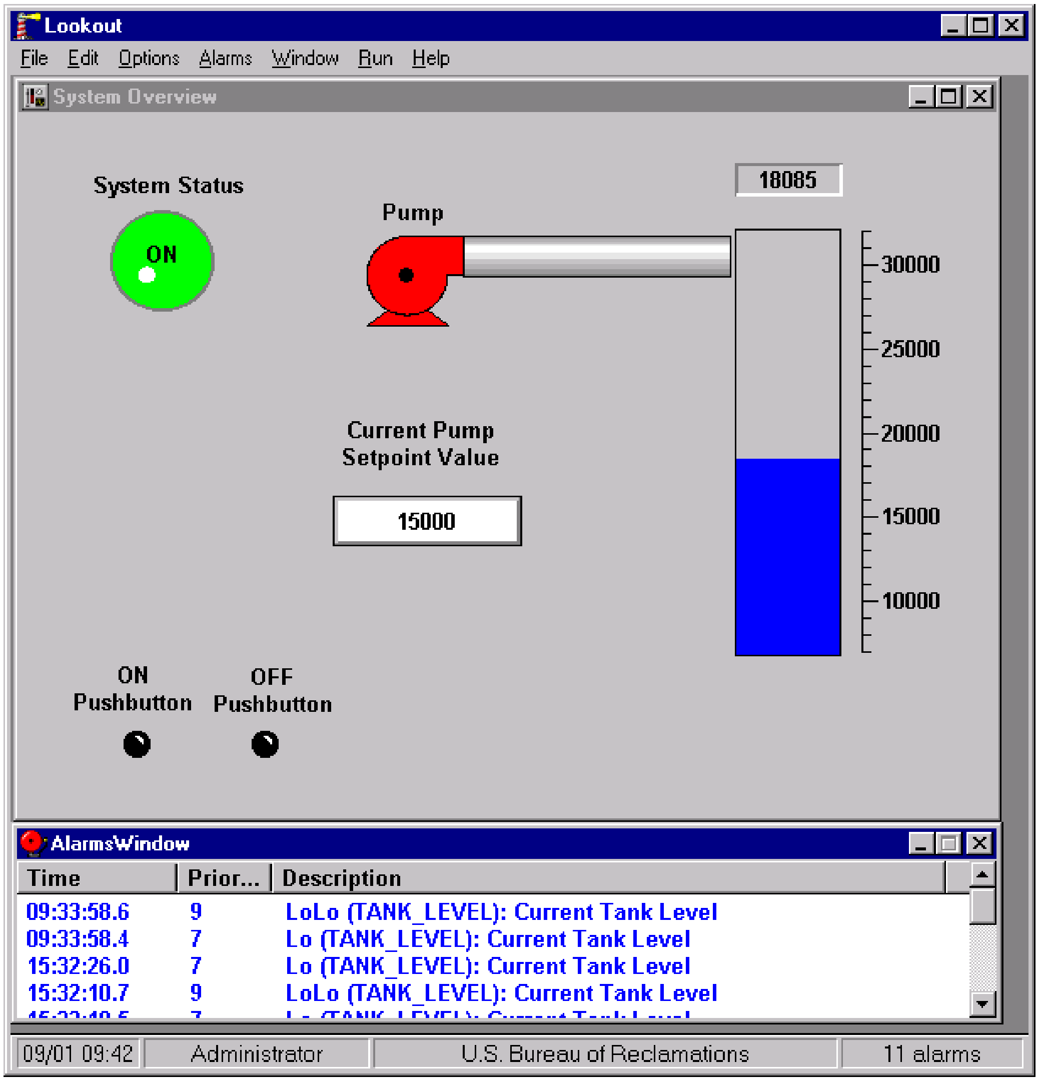
Now that each of the graphical objects have been placed, run the panel by hitting the CTRL key and Spacebar simultaneously. By clicking on the text entry box, you can change the value of the setpoint on the PLC.

Implementing Alarms:
In Lookout, the Alarms window is always visible. By default, it tracks system events such as every time a button is pressed or setpoint is modified. To implement an alarming condition on the tank level parameter:

1. In the object explorer, locate the TANK_LEVEL tag under the SCADAPack object. Right click and select Edit Database.
2. Fill in the Alarm Conditions in the dialog box:
   - Area: Tanks
   - HiHi: 30000, Priority: 9
   - Hi: 28000, Priority: 7
   - Lo: 12000, Priority: 7
   - LoLo: 10000, Priority: 9
   - Deadband: 1000
3. Select the Update button, the Quit.

Now when the TANK_LEVEL parameter surpasses one of these threshold values, an alarm will be written into the Alarm window. To suppress the events, select the Alarms menu, Filter Options and select Show Alarms Only.

This interface provided the basic functionality of monitoring and controlling the systems (See Figure 4). Of course just the bare essentials of interface development have been covered here, however these are the fundamental building blocks that Lookout offers for interface development.



Time	Prior...	Description
09:33:58.6	9	LoLo (TANK_LEVEL): Current Tank Level
09:33:58.4	7	Lo (TANK_LEVEL): Current Tank Level
15:32:26.0	7	Lo (TANK_LEVEL): Current Tank Level
15:32:10.7	9	LoLo (TANK_LEVEL): Current Tank Level

09/01 09:42 | Administrator | U.S. Bureau of Reclamations | 11 alarms

Figure 4. Final Lookout System Interface

Building the Interface using iFIX version 2.5

1.   Introduction

IFIX, (see Figure 5.) the HMI offering from Intellution, which also provides other software packages for industrial automation and asset utilization. Intellution is a division of Emerson, a global manufacturer with products, systems and services for commercial, industrial, and consumer markets.

2.   Installation

Installation of iFIX involved several steps, some of which are critical if not performed exactly correct. First, licensing of iFIX was accomplished with a hardware key or “dongle.” Because of the hardware key, there are not serial numbers to enter and keep track of with iFIX. This is a small connector that plugs into the PC’s parallel port. iFIX won’t install (or run) without the key. After the installation CD is inserted, the Autorun feature started the installation program, which was straightforward. The software in installed by default into the “C:\dynamics” folder.

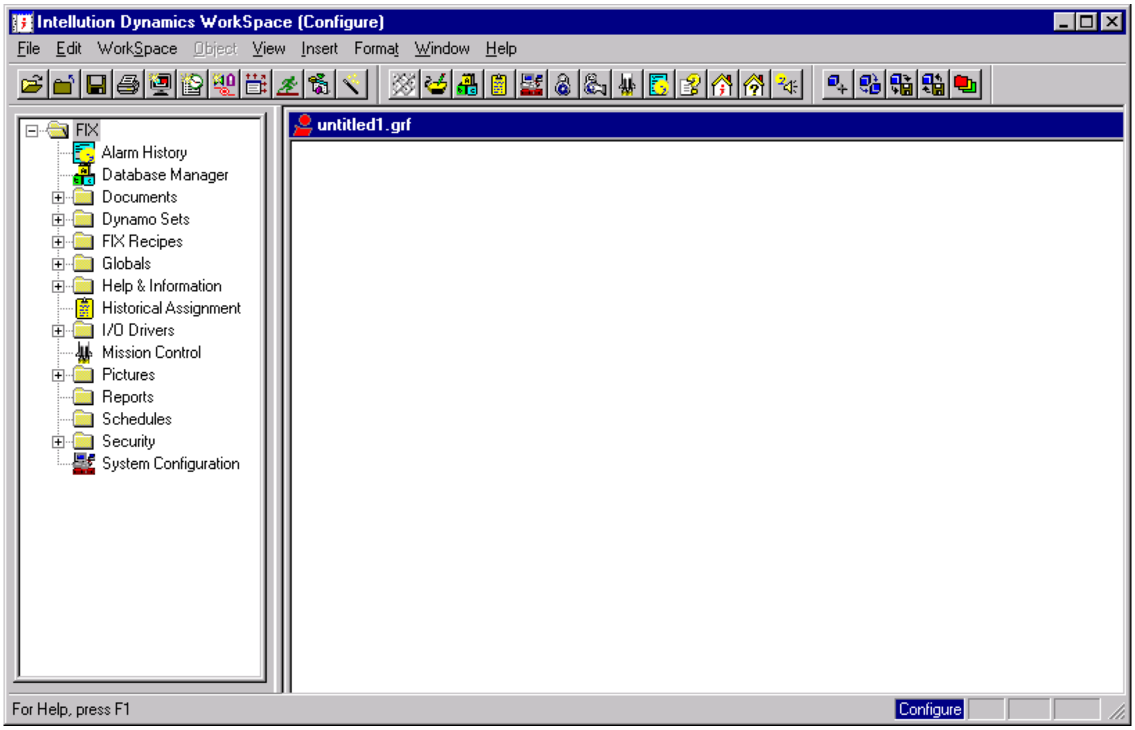


Figure 5. iFIX 2.5 Development Environment.

There was one problem encountered with the installation process, where, near the end of the process, a dialog box asked if the user would like to install any hardware drivers. At this point, I inserted the drivers CD into the drive and hit OK (It seemed the logical thing to do). This aborted the installation and it had to be performed again, in full. Up on another attempt, I answered the question “NO” and the installation performed correctly. The drivers were loaded at a later time (see next section), but nonetheless, the installation had to be performed twice before it was successful. If the Autorun feature is not enabled, the setup application on the CD is **setup.exe**.

3. Loading I/O Drivers

Loading I/O drivers in iFIX was done outside of the iFIX application. As part of the installation package, a separate CD labeled “I/O Drivers and OPC Servers” accompanied the installation CD. Simply inserting this CD started a driver installation program and allowed the user to select a driver. However, it was not readily apparent which driver was needed for the SCADAPack PLC. There were several Modbus drivers and it took nearly 2 hours and several attempts to get the correct driver installed and working.

In the end, it was the “MB1 – Modicon Modbus I/O server” driver that eventually became operational. In the process, the Intellution support web page (http://support.intellution.com) had some good resources that help to solve the problem.

However, this web site required a support contract to get full service, which I was able to obtain.

4. Building the Main Panel

Navigation and operation within iFIX was quite complicated and involved. The procedure outlined here was developed by trial and error, since no cohesive tutorial or help file seemed to layout the logical sequence of creating a simple form.

Configuring the Driver, Channel, Device and Blocks:

1. In the iFIX main panel, double click on the System Configuration Object to start the SCU (System Configuration Utility).
2. In the SCU, click on the icon that resembles a PLC to start the SCADA Configuration utility. Select the MB1 driver and the configure button.
3. On the next dialog box, select the local server and connect. This will start the Power Tool utility, used to configure the I/O driver.
4. Select the Edit menu and Add Channel.
5. Name the channel (Channel0), a description and select a Comm Port (COM 2).
6. Highlight the new channel in the object browser, and again go the Edit menu and select Add Device.
7. At this point you can verify that there is communication with the device by selecting the Display Mode -> Stats Mode. If the Transmits/Receives fields are incrementing, there is communications with the PLC.
8. Note: The iFIX help indicates that data block (tags) can be entered directly from the Power Tool utility. I was not able to get this to work. The workaround I found is outlined in the following steps.
9. Exit the Power Tool, SCADA Configuration, and SCU. Return to the iFIX main display.
10. Double click on the Database Manager, select Open The Local Node and hit OK. This is the utility used in iFIX to create tags or memory locations that can be accessed in the interface to be built.
11. Select the Blocks menu-> add. First the type of tag is selected. Select DI for Digital Input and OK.
12. Next enter the tag name (STOP), description, driver (MB1), address (Device0:10001) and a scan time of 0.01. Select Save to save the block.
13. Do the same for the other 3 digital input tags that are needed:

	Name	Address
•	START	Device0:10002
•	PUMP	Device0:00001
•	SYSTEM_STATUS	Device0:00002

14. Create an Analog Input (AI) tag for the TANK_LEVEL variable (Device0:30001).
15. Create an Analog Register (AR) tag for the SETPOINT variable (Device0:40001).
16. Once these tags are created, there current value will be displayed in the Curr Value column. There must be a Curr Value displayed before continuing.

17. Exit the Database Manager and save the database file (FIX).

At this point, the driver was configured, the tags were created and communication between iFIX and the PLC was verified. The actual development of the interface could now start.

Main Panel Creation:
All displays in iFIX are called “Pictures” and are located under a folder in the object browser. There are 4 default Pictures, which SHOULD NOT BE DELETED. When the Pictures folder was opened, an untitled1 panel was created that was saved as “Main.” This served as the main interface panel for the rest of the development.

Discrete Input Display:

1. From the Insert menu select Oval. Drag and oval onto the main panel.
2. Right click on the oval and select Animations.
3. Select the Color tab, and click on the Animate checkbox.
4. In the Data Source field, use the expression builder to select the Node Name (FIX), Tag Name (STOP) and Field Name (F_CV). Select OK.
5. In the Foreground Color section, select the Exact Match radio button, enter a color for the value of 0 and another color for the value of 1. Select OK to close the Animation dialog box.
6. Perform these steps again and create visual displays for the other discrete tags using appropriate shapes and colors: START, PUMP, SYSTEM_STATUS.
7. To run the Main panel, push CTRL-W. (CTRL-W will also return the application to Edit Mode). The displays should change according to the position of the discrete inputs.

Analog Input Display as a Digital Number:

1. From the Insert menu, select Text. Click anywhere in the Main panel and type a label (Tank_Level).
2. Right click on the text and select Animations.
3. Select the Text tab and click on the Caption Animation checkbox.
4. In the Data Source Field, use the expression builder to select the Node Name (FIX), Tag Name (TANK_LEVEL) and Field Name (F_CV). Select OK and again OK.
5. Run the panel to verify that the tank level is displayed in the text box.

Analog Input Display as a Linear Bar Display:

1. Similar to the digital indicator, draw a tall rectangle using a selection from the Insert menu.
2. Right click on the rectangle and select Animations.
3. Select the Vertical Fill field and enter a value of 0.
4. Select the Animate checkbox.
5. In the Data Source Filed, use the expression builder to select the appropriate name as before.
6. Input minimum is 7000 and maximum is 30000. Select OK.

7. Run the panel to verify that the linear bar display is working properly.
8.

Data Entry Box:

1. From the Insert menu, select Datalink. A Datalink object will be placed in the Object Browser under the Main picture.
2. Left click and drag the Datalink object to the Main panel.
3. In the Source field, select the correct filed name (FIX:SETPOINT:F_CV).
4. In the Data Entry filed, select In Place.
5. Enter the Whole digits: 8 and decimal 0. Select OK.
6. Run the panel, and test the Data entry box. Highlight the values, enter a new value and hit the return key. It will change the setpoint ok the PLC.
7.

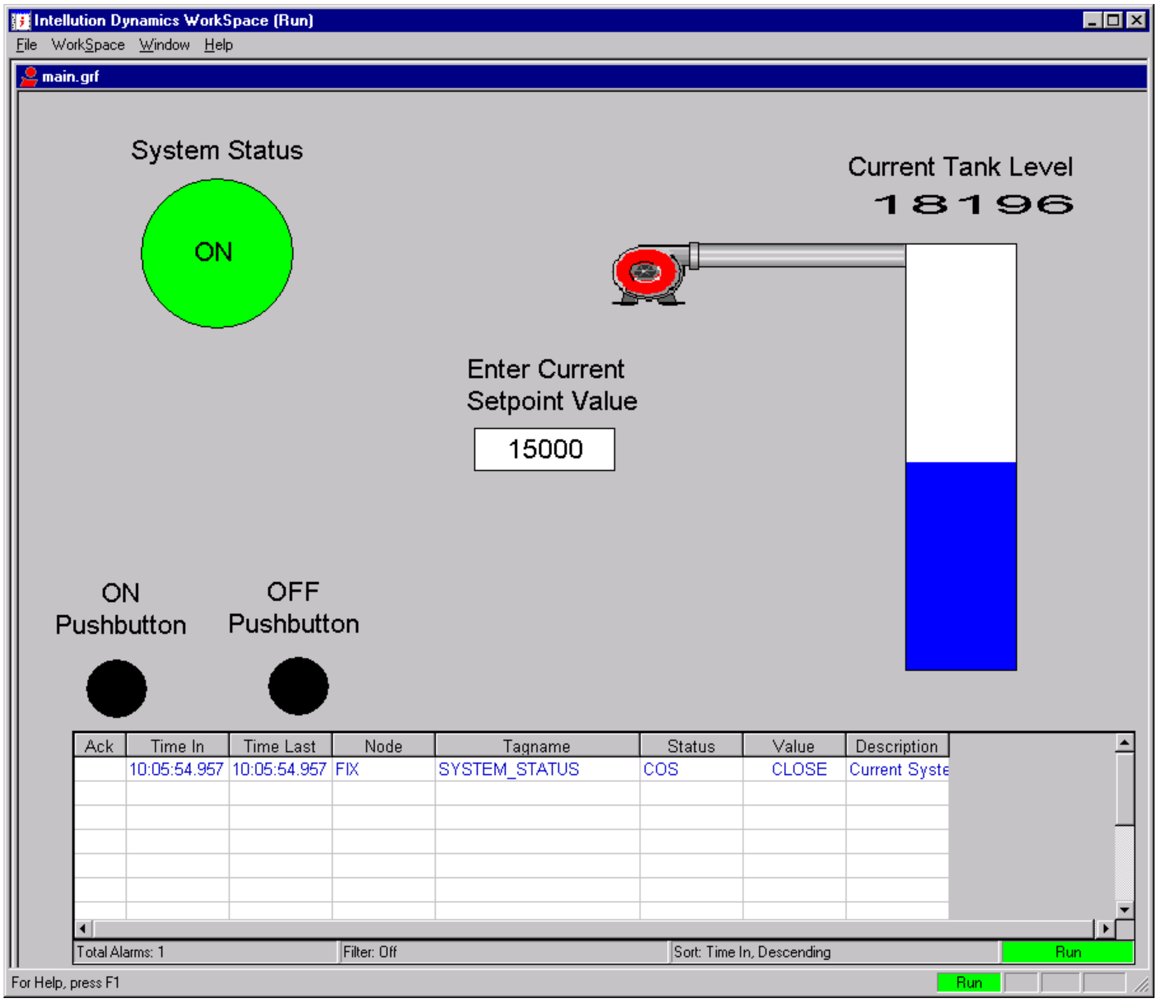
Implementing Alarms:

1. From the Insert Menu, select Alarm Summary. Drag the approximate size on the Main panel.
2. Position and size the Alarm Summary using the object's handles.
3. Now the tags that should trigger alarms must be modified. Again, open the Database Manager.
4. Double click on the TANK_LEVEL block.
5. Select the Alarms tab and enter the following values:

	Field	Value
•	Low Low	10,000
•	Low	12,000
•	Hi	28,000
•	Hi Hi	30,000
•	Deadband	1,000

6. Update the block and close the Database Manager. Run the Main panel and manipulate the TANK_LEVEL value with the potentiometer. At the high and low levels specified, an alarm will appear in the Alarm Summary.

This interface provided the basic functionality of monitoring and controlling the systems (see Figure 6). Of course just the bare essentials of interface development have been covered here, however these are the fundamental building blocks that iFIX offers for interface development.



Ack	Time In	Time Last	Node	Tagname	Status	Value	Description
	10:05:54.957	10:05:54.957	FIX	SYSTEM_STATUS	COS	CLOSE	Current Syste

Figure 6. Final iFIX System Interface

Evaluation:

The ideal of this project was to establish an objective evaluation of the two HMI packages, Lookout and iFIX. Of course, with the un-formalized approach of this project and the fact that a single person did all of the development and evaluation tasks, it was impossible to give a truly unbiased representation of these two packages. There were many extraneous factors to consider such as: my previous experience with similar HMI software packages, which package I evaluated first, how much documentation of each package I read, etc. However, every attempt was made to be as objective as possible while performing these evaluations and I summarized each of their strengths and weaknesses as truthfully as I could.

The format of this evaluation was to identify topics that I feel were important to a user considering each of these packages. I drew from my background working with many different hardware and software combinations and the struggles I've experienced with similar systems.

Lookout	**iFIX**
Installation	
The installation of Lookout was more straightforward. It took less than 5 minutes and there were no questions that were ambiguous or difficult to understand. It placed the files in the C:\Program Files\ folder, which is a Windows standard practice and is better for an installation where many applications will be loaded.	The installation of iFIX started very well, but aborted when I inserted a driver CD that I thought it was asking for. It turned out that I should have waited and installed the driver after the initial iFIX install, but the installation instructions were quite ambiguous and confusing. Total time was approximately 1 hour to install.
Advantage: Lookout	
Licensing	
From a user's perspective, "soft" licensing is much more desirable than using a hardware key. All that is needed is the software's serial number and the installation CD. Upgrading to "higher" versions of the software (with more I/O points) is accomplished with the entering of the serial number into the software.	iFIX uses a hardware key, which is often troublesome to keep track of, and makes moving the software to another PC very difficult. In addition, upgrading the software to more I/O points requires a new hardware key. Software licensing technology is to the point that all companies should be using it and discontinue hardware key licensing.
Advantage: Lookout	
Loading I/O drivers	
Lookout Modbus driver installation and configuration was all accomplished within the main Lookout application. The one disadvantage was that a control panel was first required before a driver could be loaded. Overall, the process was intuitive and easy to perform.	iFIX Driver installation was confusing, as it was not known which driver was needed for the SCADAPack PLC. It took several hours to resolve. Additionally, the driver installation and configuration required the use of several external applications, which very not intuitive and required much investigation before the driver was operational. This was a very difficult aspect of the overall development process.
Advantage: Lookout	

Tag Creation

Tag creation in Lookout was simple and intuitive. The addressing scheme matched closely with Modbus nomenclature. Creation, deletion and modification were all accomplished with a single interface. Alarm conditions were also specified within the same dialog box.

This is one of the weakest aspects of iFIX. First, it is necessary to start an external application to add and modify tags. Second, the addressing of the device-memory location was extremely confusing and it was just by trial and error that it was finally successfully configured. One major discontinuity is that the name of a tag configured in the Power Tool application has no correlation to the name of a tag configured in the Database Manager. This is an example of very poor integration from a human factors perspective.

Advantage: Lookout

Creating Displayable Objects

Visual object creation in Lookout is performed by inserting the tag data type first, followed by the visual representation. This is a good approach to the process, and works well for differing types of tags. The library of pre-configured objects is decent and their use is easy to implement. Objects such as text entry boxes are fast and easy to use.

The basic shape creation functionality in iFIX was very good. The graphics tools were complete and easy to use. Animation was easily added to basic shapes and this process was especially intuitive and easy to accomplish. iFIX includes several pre-configured visual objects, which are called Dynamos (tanks, pumps, etc.). The selections of Dynamos are very limited, but a user can create and use their own. More sophisticated objects, such as text entry boxes are not as intuitive and require help files and trial and error to configure correctly.

Advantage: iFIX

Overall Development Functionality

Overall, the Lookout interface is simple and easy to use. The learning curve is quite short, and in a short time, the user can create simple interface panels. More complex interfaces and functionality are not included in the basic setup. If more sophisticated objects are desired, extensive customization (using tools such as Visual Basic) would be required.

Overall, the iFIX interface is complex and difficult to learn. Due to the lack of a good interface overview, it takes time to understand the iFIX approach many of the required tasks (driver, I/O configuration, tag creation, etc.). With many of the separate applications (i.e., Power Tool, Database Manager), it seems to be a collection of several applications instead of a cohesive, integrated solution. It appears to have tools in place for more complex customization, however it is anticipated the need to jump into development tools, such as Visual Basic, would still be required.

Advantage: Lookout

Help files, Support and Tutorials

Lookout contains a standard, searchable Windows help file, which is neither especially good nor bad. The real advantage was the Lookout tutorial that walked the user through basic functionality in about an hour. Time spent with this tutorial was returned many times over during the interface development for this project.

iFIX contains a standard, searchable Windows help file in addition to several Electronic Books on a wide variety of subjects. However, the indexing and working of the help topics never really seemed to give the specific answers that were needed. The lack of a tutorial was extremely disappointing, as the functionality just about every aspect of iFIX had to be accomplished by a trial and error methodology. A good example is how to load and configure a driver. The help file referred the user to the Power Tool help file, which gave very detailed descriptions of the Power Tool’s user interface, but no information on how to load, configure or address memory locations on the PLC. It just took a long time to figure out how to perform basic functions in iFIX.

Advantage: Lookout

Overall Ease of use

By now it should be very apparent that Lookout version 4.5 was far superior with regard to ease of use. The entire application is a tightly integrated, consistent, coherent and guessable interface. The basic tutorial is somewhat responsible for this, as it introduces the core concepts and terminology, making it easier to find specific answer in the help file as the user learns the entire application. This is a quality software product.

iFIX version 2.5 leaves much to be desired in terms of usability. In fact, it is difficult to envision a new user, without any HMI experience, making much progress without some kind of a training experience (perhaps by Intellution's design!). The lack of consistency in the interface and the poor quality of the help were especially frustrating. Intellution has quite a bit of work ahead if they are going to improve the usability of iFIX.

Advantage: Lookout

Summary:

The objective of this project was to provide ITRC with a baseline of current commercial HMI packages in terms of software usability. In general, it was an attempt to expand the value-added contribution that ITRC makes in the regional irrigation modernization industry. It is desirable that, in the future, this analysis method could be expanded to include more of the HMI vendors that offer software for this growing industry.

By approaching this project from a realistic situation (using an automated water level control system), a near real-world experience was encountered. From this experience, it is proposed that an objective evaluation was performed. While both packages resulted in identical interfaces, Lookout clearly out-performed iFIX in nearly all of the areas considered. While the data gathered in this project suggests Lookout to be the superior HMI package, of course, a final decision would be left up to each irrigation district, as other factors my be considered in the final HMI software selection.

Bibliography

Intellution (2001). iFIX Windows Help File. Included as part of the iFIX 2.5 installation CD.

Intellution Support Web Site (2001). http://support.intellution.com/ Viewed 20 August, 2001.

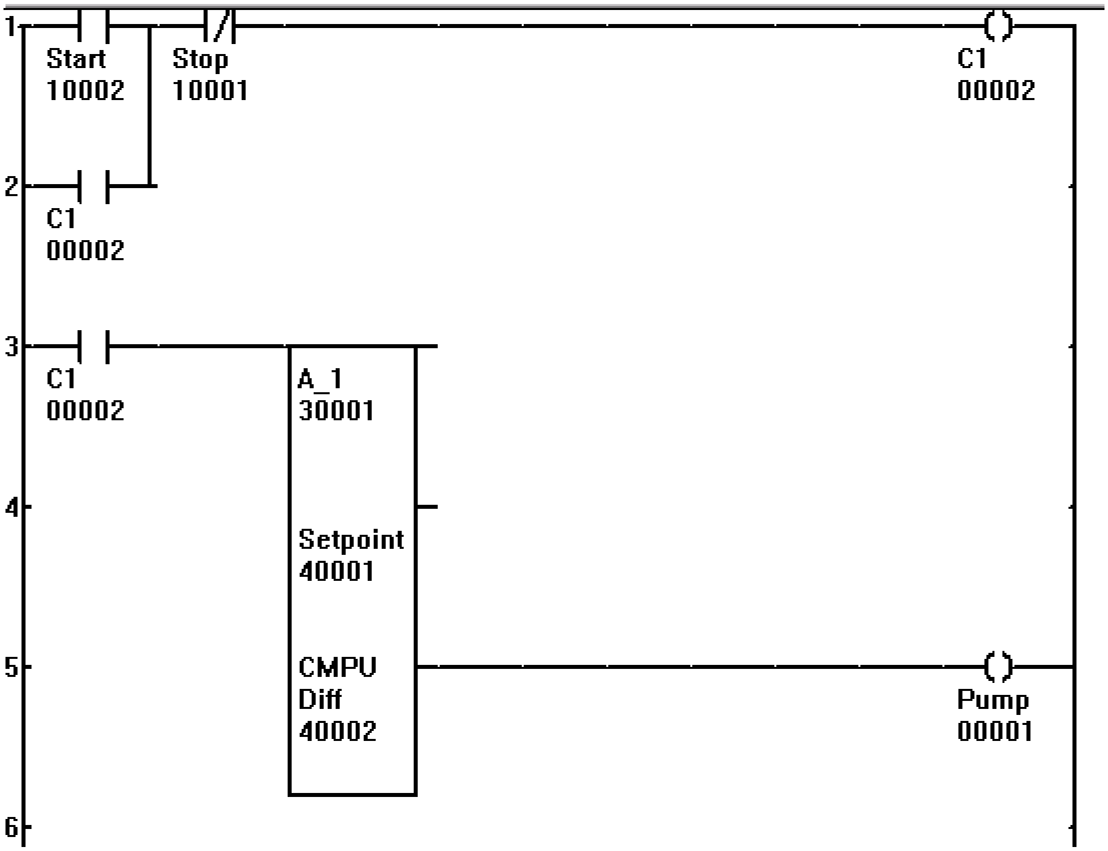
National Instruments (2001). Lookout Evaluation Guide. Included as part of the Lookout 4.2 installation CD. Chapter 3: *Try Out Lookout.*

National Instruments (2001). Lookout Windows Help File. Included as part of the Lookout 4.2 installation CD.

National Instruments Support Web Site (2001). http://www.ni.com/support/ Viewed 25 August, 2001.

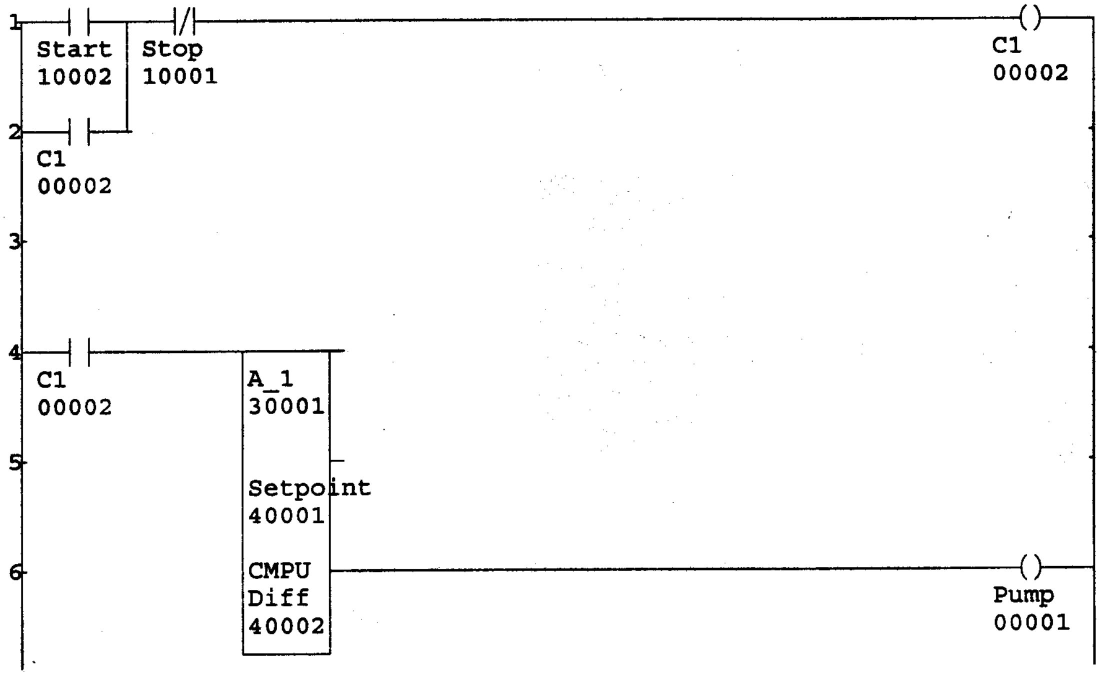
Waters, Robert M. (1994). User of Standards to Reduce Human Error. SERA-Vol. 2, Safety Engineering and Risk Analysis. ASME. PP 161-166.

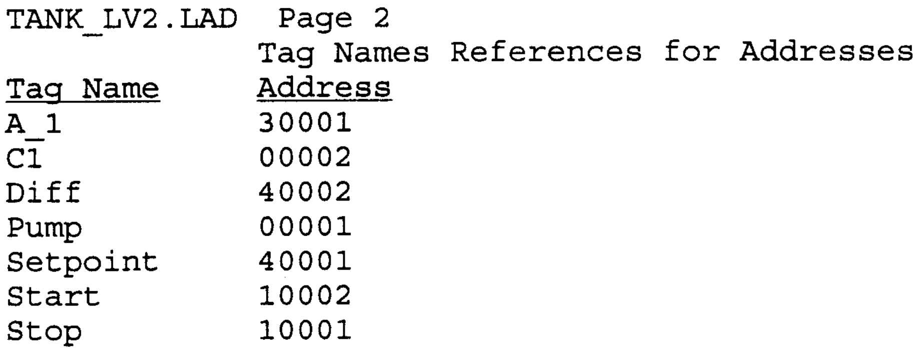
Appendix A: Ladder Logic Documentation



TANK_LV2.LAD Page 1 Ladder Logic Network

network 1:

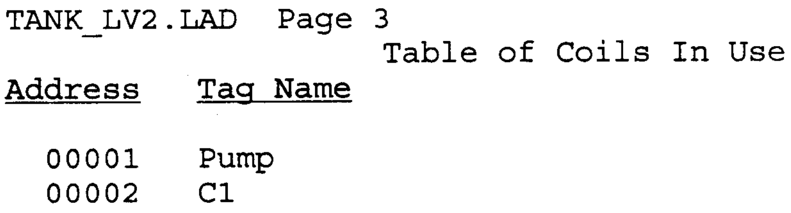




TANK_LV2.LAD Page 2

Tag Names References for Addresses

Tag Name	Address
A_1	30001
C1	00002
Diff	40002
Pump	00001
Setpoint	40001
Start	10002
Stop	10001



TANK_LV2.LAD Page 3

Table of Coils In Use

Address	Tag Name
00001	Pump
00002	C1