The Sage Advisor

SCADA, SECURITY & AUTOMATION NEWSLETTER

Volume 29, Issue 2 • Fall 2019

A Publication of Sage Designs, Inc.

Climbing Over Ladders and Functioning with Blocks: Program Your Way

There are the 5 languages making up the open international standard (IEC6-1131-3) for Programmable Logic Controllers (called PLCs, PACs, or rPACs).

With this rich programming selection, how does one go about making the right choice about which to use.

There's no simple answer to the question, but in this article we will take you through a few differences between three of these languages, as well as some of the advantages each offers, so that you can make the right decision for your system.

There are three graphical languages; Function Block Diagramming (FBD), Ladder Diagramming (LD), and Sequential Flow Charts (SFC). There are also two textual languages; Structured Text (ST) and Instruction List (IL). Sequential Flow Charts and Instruction List are rarely used, so for the purposes of this article we will discuss three languages which make up the overwhelming majority of programming currently in the field: Ladder Diagramming, Function Blocks, and Structured Text.

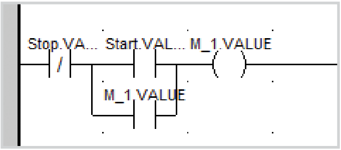
Ladder Diagramming is an obvious choice and easy choice for many industry professionals because of its prevalence in the United States. An estimated 80% of the PLC code in use today (within the US) is written in Ladders, making Ladders an important tool for the maintenance and operation of existing systems. However, it is important to note that while Ladders has a clear dominance in existing systems, 80% of new code being written today is written in Function Blocks. And while Ladders enjoys dominance in the US, Function Block and Structured Text are substantially more prevalent in the global market. But let's center back onto our core question: What makes sense for you?

One commonly raised point is "All our existing code is in Ladders. It's what our technicians and engineers know how to use". Which is a possible advantage of Ladders for those long-established SCADA systems. For many professionals the familiar rails and rungs of Ladder Diagraming (LD) are familiar and offer a short learning curve. The ability to transfer Ladders programs from obsolete equipment to new requires less effort than rewriting in a new language, which can shorten development time. Plus, when you have an entire shop used to working in Ladders you will usually need to overcome resistance from the parts of your team who have been used to doing the same thing the same way for years to introduce something new like Function Blocks or Structured Text.

While the "this is what we do" argument is compelling in the short term, it has less relevance in the long term. As the industry here and around the world gravitates towards modern languages with their additional benefits and powerful new tools, any technician or engineer that wants to keep their skill set up to date had better take a look at programming in more than one of these. While a control task can be accomplished in any of these languages each has its place and a smart programmer will mix and match to make the best use of their strengths.

Let's take a look at doing a couple of things in each of these to show an example of where each language might be put to best use. For starters, let's consider a simple starter motor stop/start program. The first one I've chosen to display (figure 1) is built the way most Electricians prefer to see it, with the Stop button proceeding the rest of the elements to match how it's wired in the field.

Figure 1



There is no more obvious way to write this piece of code. This is clear and recognizable to anyone who has worked as an industrial electrician.

Climbing Over Ladders, *Continued on Page 6*

Pillbox Enclosures Installed Along Major California Canal

The newest Pillbox installation sites along a major California canal system have been installed with the goal of adding instrumentation to address concerns of subsidence in the valley.

According to the USGS, continued groundwater-level and land-subsidence monitoring in the San Joaquin Valley is warranted because groundwater levels are poised to decline when surface-water deliveries do not meet demand, which may result in additional land subsidence. Even in precipitation record-setting years such as 2010-11, water deliveries fell short of requests in the Central Valley. Therefore, it is likely that groundwater levels will decline in the future. Integrating subsidence, deformation, and water-level measurements—particularly continuous measurements—permits analysis of aquifer-system response, which enables identification of the preconsolidation head and calculation of aquifer-system storage properties. This information could be used to improve numerical models of groundwater flow and aquifer-system compaction, to refine estimates of governing parameters, and to predict potential aquifer-system compaction which could be used to manage water resources while considering land subsidence.

As a part of this push for more data, the canal authority is adding subsidence monitoring equipment along a large area. The need for ruggedized, vandal resistant installations which can be installed with a minimum amount of disturbance to this ecologically sensitive area, they chose to install several PillBox enclosures along the canal. According to the Project Engineer, they have level instrumentation all along the canal as there's subsidence happening all throughout central and San Joaquin California. The ground is lowering, in terms of ground elevation, which makes it really tough to move water along the state. The new level sensors are going help manage flow better, level, so they actually don't over-top the edges of the canal or run too low where some other event could occur.

Among the reasons to choose the Pillbox specifically for this application were that they really like the low profile installation of this Pillbox, as well as some of the hardened design for anti-theft, tamper-proof, and then just ruggedness of the solution. They also have environmental concerns along the aqueduct that make it really tough to get permitting to install larger type of equipment.

This installation proved to be significantly less expensive than traditional pad-mounted enclosures and masted equipment, which is a primary advantage to any Pillbox installation.

CELEBRATING 25 Years ANNIVERSARY

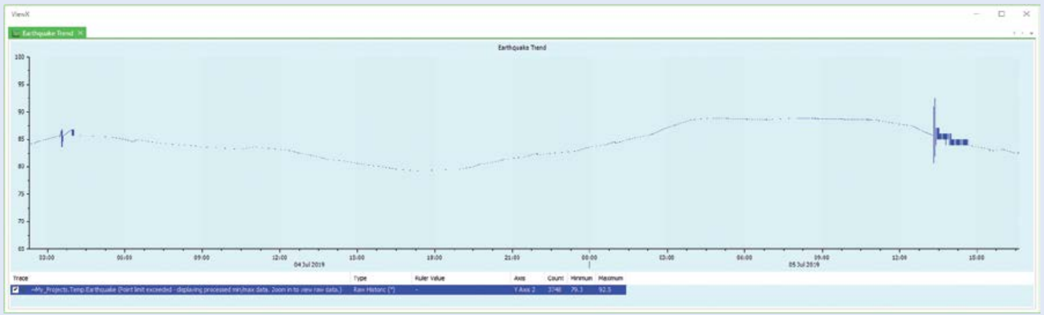
SCADAWise PillBOX

Inside this issue:

Earthquakes vs. SCADA: What's Shakin'

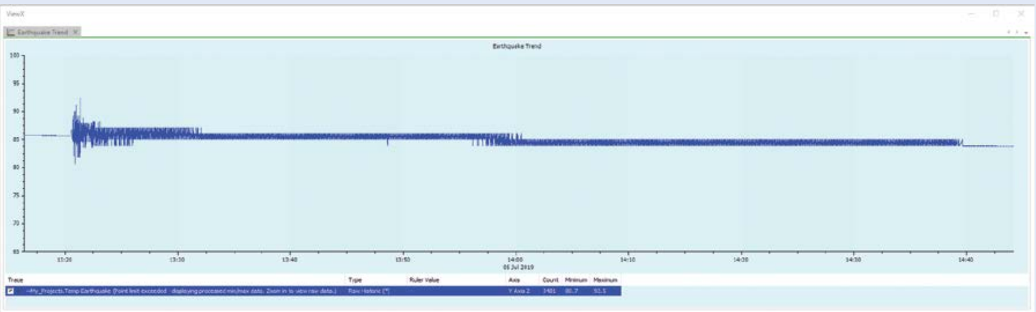
On July 4th and 5th this year, the city of Ridgecrest California experienced two significant earthquakes and put everyone within 100 miles of a fault line on notice that we need to plan for such events in everything we do, especially when it comes to our water system and other infrastructure. For everyone who hasn't been physically present at a tank during a quake we can now show you through the tools available in a modern SCADA system the dramatic effects of an earthquake on tank water levels.

Note: These trend charts in this article are from a tank over 80 miles away from the epicenter, clearly showing why we harden our systems from earthquakes the way we do.



Trace	Type	Ruler Value	Axis	Count	Minimum	Maximum
☑ ~My_Projects.Temp Earthquake (Point limit exceeded - displaying processed min/max data. Zoom in to view raw data.)	Raw Historic (*)	-	Y Axis 2	3748	79.3	92.5

Chart 1



Trace	Type	Ruler Value	Axis	Count	Minimum	Maximum
☑ ~My_Projects.Temp Earthquake (Point limit exceeded - displaying processed min/max data. Zoom in to view raw data.)	Raw Historic (*)		Y Axis 2	1401	80.7	92.5

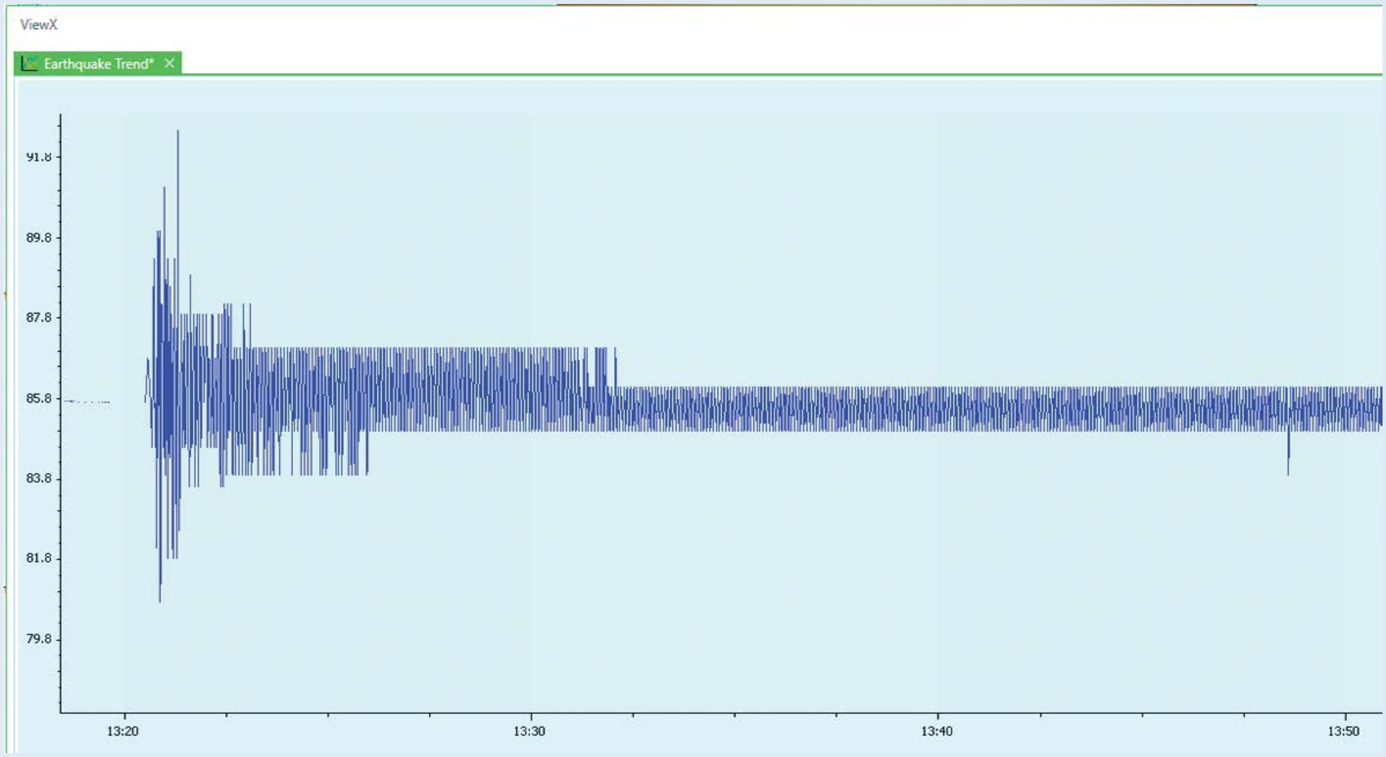
Chart 2

The first chart shows both events in coarse detail. The tank here is about 35' high and the level signal is scaled from empty to 100% of the safe fill height. During the event you can plainly see the level, which was at 82% just prior to the event, swing from 80.7 to 92.5%; **a change of more than 4 feet.** The data was collected in high resolution because this water district uses DNP3 Secure Authentication protocol from x70 SCADAPacks back to a ClearSCADA host so that the data is collected and time-stamped at the controller and sent up to the host in batches as time allows. The result is data in the fractional second rate which is transmitted over a narrow-band radio as time allows.

The second chart shows the 7.1 earthquake in greater detail, and you can see the minutes-long event disturbing the tank level for more than an hour. This clearly demonstrates why programming of the controller responsible for maintaining the tank level must take unusual events into consideration so that pumps will operate in a predictable way under all circumstances.

As our SCADA systems become more and more powerful, they begin to reveal interesting facts about how our systems act and react. Not just for earthquakes but for water hammer or other pressure spikes. Can your system record system-wide events and report them in sufficient detail to allow for better design? Have you prepared your system for a tempest in a tank?

A DNP-based system may reveal things about your system that you did not know. Let us know if you are interested in a demonstration of a DNP3-based SCADA system.



Detail of First 30 Minutes

SAGE DESIGNS, INC.
SCADA & Security Products

Earn Contact Hours

Schneider Electric

SCADAWise™ Training Classes

ClearSCADA

ClearSCADA Level 1 Training Course

March 23-26, 2020 — Mill Valley, CA
May 11-14, 2020 — Buena Park, CA

Day 1 (8AM - 4PM) Installing ClearSCADA, Introduction to ClearSCADA, Components, Using ViewX, Using WebX, ClearSCADA Help

Day 2 (8AM - 4PM) Configuring using ViewX, Database Organization, Basic Telemetry Configuration, Creating Mimics, Creating Trends

Day 3 (8AM - 4PM) Configuring using ViewX, Templates & Instances, Logic Languages, Security, Communications Diagnostics

Day 4 (8AM - 4PM) Reports, System Configuration, System Architecture, Questions

Cost: ClearSCADA Training Course $2,200

Sage Designs' ClearSCADA Level 1 Course has been certified by (a) the California Department of Public Health as courses qualifying for contact hour credit for Water Operator Certification for Drinking Water Treatment or Distribution in the State of California and (b) the State of Nevada Department of Environmental Protection, Bureau of Drinking Water for contact hours towards the Nevada Drinking Water Operator Certification Program.

(28 Contact Hours)

ClearSCADA Level 2 Training Course

September 23-25, 2019

Day 1 (8AM - 4PM) Installation, Understanding the Architecture of ClearSCADA, Application Design Considerations, Server Automation Interface, ClearSCADA Logic Engine, Using ODBC and SQL.

Day 2 (8AM - 4PM) Advanced Mimic Design and Techniques, Data Grids and Data Tables.

Day 3 (8AM - 1PM) Accessing Historical Data, Ad Hoc trends, Archiving

Prerequisite: ClearSCADA Level 1 Training Course

Cost: ClearSCADA Level 2 Training Course $1,825

SCADAPack

Telepace Studio Training Course

March 10-11, 2020 — Mill Valley, CA
May 5-6, 2020 — Buena Park, CA

Day 1 (8AM - 4PM) SCADAPack controller operation, Series 5000 I/O, Telepace Studio introduction

Day 2 (8AM - 4PM) Telepace Studio advanced programming techniques and advanced functions

Day 3 (8AM - 2PM) Controller communications, Modbus Master/Slave protocol, Diagnostics, Modems

Cost: SCADAPack Telepace Studio Course $1,650*

* You must have a licensed copy of Telepace Studio installed on your computer for this course. If you do not have a licensed copy, you may purchase one with the class at a special course price. Course price for Telepace Studio: $510 + applicable CA sales taxes

Sage Designs' Telepace Studio Course has been certified by (a) the California Department of Public Health as courses qualifying for contact hour credit for Water Operator Certification for Drinking Water Treatment or Distribution in the State of California and (b) the State of Nevada Department of Environmental Protection, Bureau of Drinking Water for contact hours towards the Nevada Drinking Water Operator Certification Program.

(14 Contact Hours)

Remote Connect & SCADAPack x70 Logic Programming

November 19-21, 2019 — Buena Park, CA
April 21-23, 2020 — Corte Madera, CA

Course Description: This three-day hands-on course is designed to give each participant a detailed introduction to the SCADAPack x70 controller series (SCADAPack 47x and 57x) and RemoteConnect, its configuration and programming tool. Topics include system configuration using RemoteConnect to configure communications protocols, point addressing, and SCADAPack x70 variable types, introduction to the Unity based logic editor, and the use of scanners and function blocks to access remote data.

Cost: Remote Connect & SCADAPack x70 Logic Programming $1,800

Schedule Your Own

ClearSCADA Test Drive

Free Hands-On Test Drive

Call to Schedule a Test Drive

Call 1-888-ASK-SAGE
email: info@scadawise.com

SAGE DESIGNS, INC.
SCADA & Security Products

Download the Registration form at: http://www.SCADAWise.com

*** * * Registration Deadline: 4 weeks before 1st day of course * * ***

All registrations are subject to cancellation fees. A confirmation notice will be sent to all registrants on or before the deadline date.

New ClearSCADA Features in 2017 R3

ClearSCADA 2017 R2 delivered a new integration experience with the SCADAPack 570/575 RTUs. The RTUs are the first models that are able to share programs with Schneider Electric Modicon M340 and M580 PACs (Programmable Automation Controllers).

ClearSCADA 2017 R3 builds on that enhanced functionality by providing the following:

- Support for SCADAPack x70 system data
- Support for routing tables:
  - » IP Routing Table
  - » DNP3 Routing Table
- Support for an IP Whitelist that you can use to control the flow of network traffic to and from a SCADAPack x70 device
- Firmware download capability, available as a 'Download Firmware' pick action
  - » A new SCADAPack x70 Firmware Image item is available and is linked via a field on the Device Configuration item for the SCADAPack x70 device.
- Alarm limits tuning on protocol-specific analog and counter points
- A new SCADAPack x70 Security Configuration item that can be used to import a Security Configuration file
- DNP3 Pulse Actions.

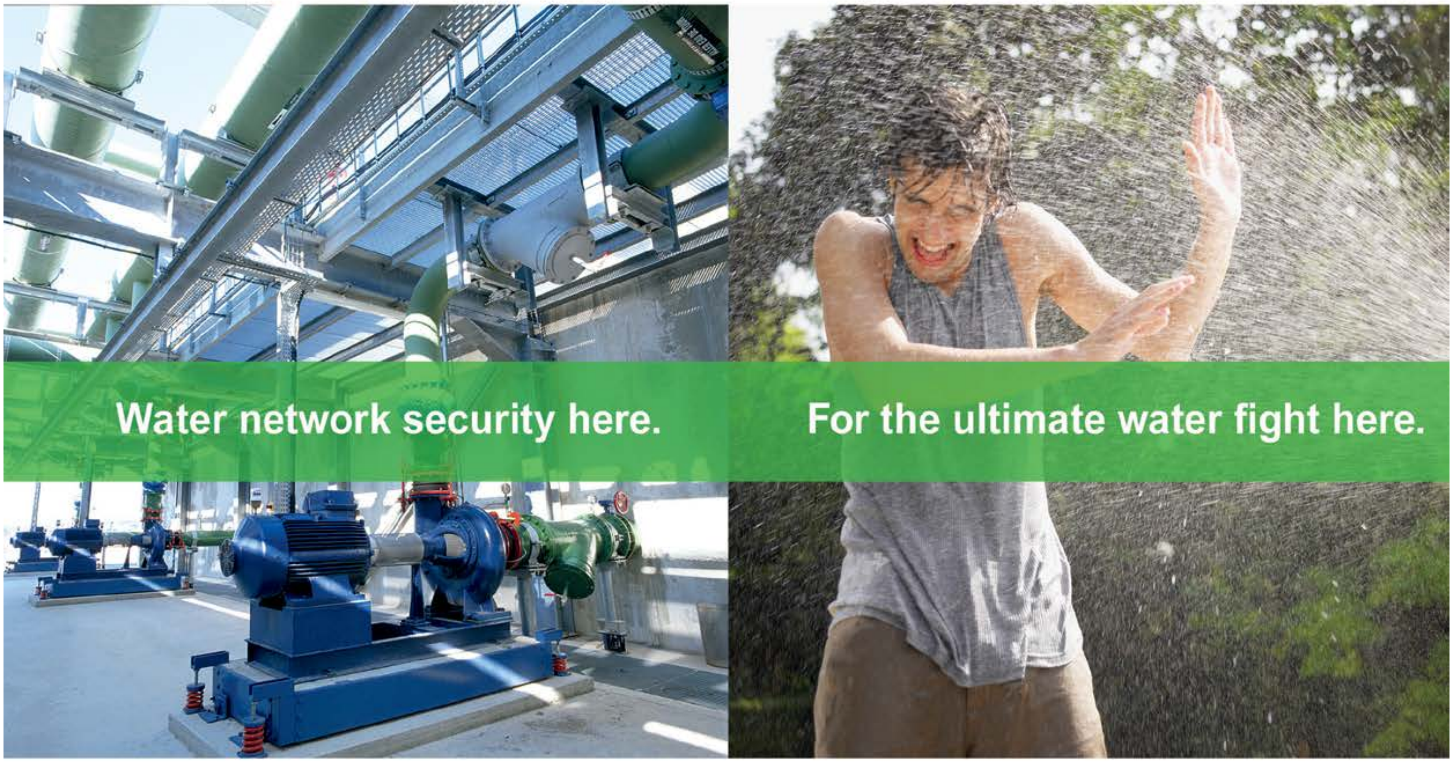
Other enhancements include:

- The DNP3 and Modbus tabs on SCADAPack x70 Analog, Digital, and Counter Configuration items are now optional
  - » Validation errors for SCADAPack x70 configurations with conflicting address/register assignments are only raised if the tab for the associated communications protocol (DNP3 or Modbus) is enabled.
- SCADAPack x70 logic programs are validated against the ClearSCADA configuration to confirm they are compatible
- ClearSCADA now uploads logs from SCADAPack x70 devices following a configuration download. These can be viewed using a pick action on the protocol-specific outstation that is associated with the SCADAPack x70 device. Additionally, ClearSCADA will raise an alarm if it detects errors in the log file. The errors themselves are logged in the Event Journal, and can be viewed using the Events List.
- You can now remove the local copy of the SCADAPack x70 logic application that has been imported into ClearSCADA. You can do this using a 'Remove Logic Application' pick action on the SCADAPack x70 Device Configuration item.

If you are a current ClearSCADA support program member, more information about these updates and improvements can be found in your updated Help file. If you'd like a more personal conversation, or are outside a current support contract, please contact your local Sage Designs representative for more information!

BIG NEWS: As a part of Schneider Electric's commitment to continued improvement and development of ClearSCADA, ClearSCADA will be joining other Schneider offerings in the EcoStruxure software family. Look for ClearSCADA under its new EcoStruxure name as EcoStruxure GeoSCADA Expert starting in 2020!

Life Is On | Schneider Electric


Water network security here.
For the ultimate water fight here.

Whatever the size and complexity, ensuring safe, reliable and secure operation from field to enterprise

Flexible I/O configuration to handle expansions
Reduce engineering time with replicated object templates

SCADA Components:

- SCADA Expert ClearSCADA management software
- SCADAPack Smart RTUs & rPACs
- Accutech wireless instrumentation
- Trio data radios

For more information, live equipment demonstrations, a listing of contact-hour certified classes near you, or to schedule a consultation on solving your SCADA challenges, contact your local Schneider Electric | TRSS representative:

SAGE DESIGNS, INC.
SCADA & Security Products

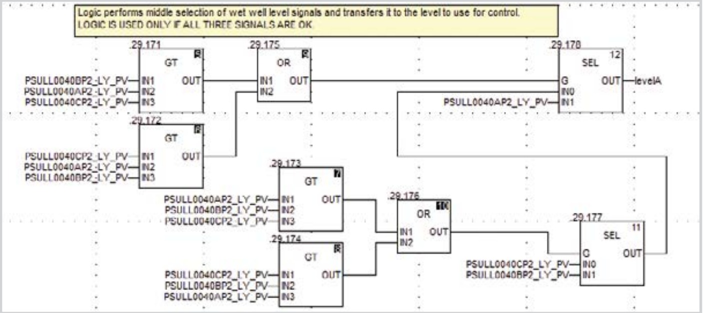
(888) 275-7243
www.sagedesignsinc.com
www.scadawise.com

SCADAWise Training

Climbing Over Ladders and Functioning with Blocks: Program Your Way, *continued from page 1, continue on page 7*

For straightforward logic, Ladders Diagramming works very well. Simple logic was the impetus for the development of the Ladders language more than a half century ago and it has proven to be a good tool for that for a very long time. At this point a Ladders programmer might be wondering "so why use anything else?" but, aside from the concern of staying current in the industry, that would leave on the table the many additional features and solutions offered by the other prominent languages.

Let's take Function Block Diagramming for our next example. It looks different, even missing the contacts and coils that Ladders programmers are used to seeing. If you look at a function block program (figure 2), you can see right away the flow of the program as one block outputs values to another through a connection line. This visual "map" of the program makes the environment an intuitive way or creating code for control.



Logic performs middle selection of wet well level signals and transfers it to the level to use for control.
LOGIC IS USED ONLY IF ALL THREE SIGNALS ARE OK

Figure 2

Recall the example code used previously (figure 1). It's a clean and simple piece of code. However, you can use Function Blocks to accomplish the same task by creating a custom function block made from that (figure 1) code. This gives you a simple way to use this code in a program without having to mix and match your languages. In the next example (figure 3) I have written the code in Ladders but more the way a programmer would like to see it.

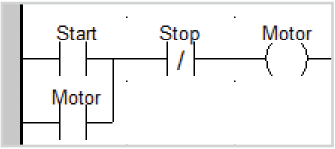


Figure 3

The result of this code is functionally the same as figure 1 but eliminates the need for one vertical shunt. I can then create a custom function block (figure 4) from the code in either figure 1 or figure 2.

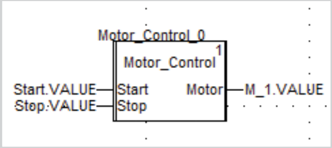


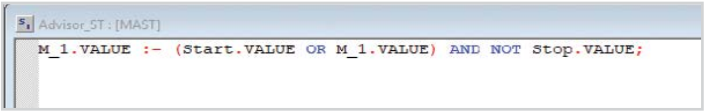
Figure 4

Once added to the library, this custom function block (figure 4) can be dropped into any Function Block program and used as many times as you like by just connecting the inputs and outputs from the rPAC's IO as needed.

We have just created a custom function block which reduces Ladders code to a simpler form and, without losing any functionality, gained clarity. Similarly, much more complex function blocks can be created from Ladders, Structured Text or in Function Block Diagraming itself. These custom function blocks make it simple to reuse your code (we will address that more in this article). Another nice feature of Function Blocks is that you can easily change the order of execution of the code which is, if you've ever faced the nightmare that is an execution order error in Ladders, a true blessing.

Our third language, Structured Text, while it is not used as widely as either FBD or LD is truly more of a "language" than the languages discussed so far. While a bit intimidating to some, properly done, it can be clear and readable. Structured Text also has the strong advantage that it can support complex math with a typed-in formula without the cumbersome functions required by LD and FBD languages.

When using Structured Text, all of the graphical elements of Function Blocks and Ladders go out the window. However, as we mentioned earlier, with this new format comes a native processing of complex mathematics and formulas via a translatable pure-text medium. For example, the same code we created in Ladders and Function Blocks created in Structured Text might look like this (figure 5):



```
M_1.VALUE := (Start.VALUE OR M_1.VALUE) AND NOT Stop.VALUE;
```

Figure 5

While a powerful tool, Structured Text has been a bridge too far for some programmers with its significant changes in interface and methodology from the majority of field-deployed PLCs. It can be further hampered by difficulty with the complex and layered equations for more complicated functions than the simple starter. As we say: Having the right tool for everything usually takes more than one tool.

Now that we have a basic awareness of the three languages, let's go deeper into their contrasts. In Unity, and the SCADAPack X70 logic compatible with Unity, each of these languages have the same library of built in functions. In each of these you can simply insert a function by browsing the Function Library for the type you need and adding it to your program section. The main difference between the representation in LD vs. FBD is that there is one enable input (EN) which needs to be active for the block to work and one Enable Output (ENO) point for connecting to the EN of the next block in line. The EN and ENO are not necessary but can also be activated for use in the Function Block programs if needed. With ST, you get a line of code for each of the inputs and outputs of a function similar to the set in the FBD case.

For an example of this difference, we will compare each of the three languages while converting Degrees Fahrenheit to Degrees Centigrade.

This is a simple program in LD where you must do adding, subtracting, multiplying and dividing in separate blocks (figure 6). For ease of comprehension I've combined the functions of dividing by 9 and multiplying by 5 into dividing by 1.8:

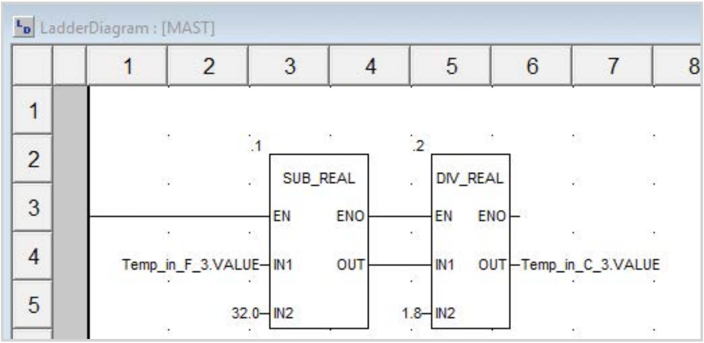


Figure 6

Most SCADA technicians should have little problem working out how this works. Below in figure 7 is what the program looks like while being monitored online in the Unity for SCADAPack software:

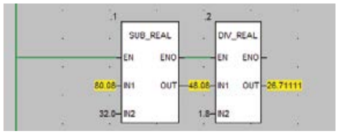


Figure 7

While FBD programs look similar to LD, there is no need to connect up an enable (EN) input to the blocks, the workspace for the functions is not constrained to the area between the L1 and Neutral rails, and the format is free-form so blocks can be moved around on the screen. If desired, the EN and ENO connections can be enabled in Function Block programming. Once enabled, they work the same way they do in Ladders.

Below is the temperature conversion written in Function Blocks (figure 8). While different, with a little thinking, this at least as easy to understand as the example in Ladders.

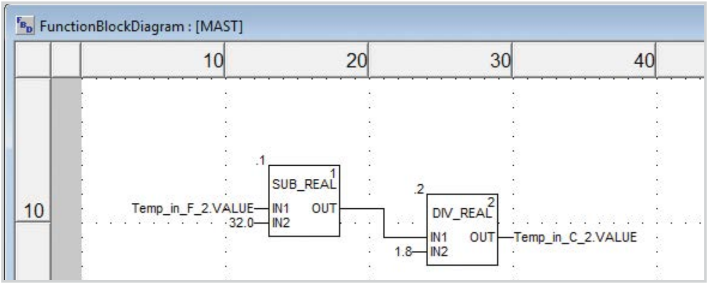


Figure 8

As you can see, the blocks bear the same name and perform the same function. See figure 9 for an image of the FBD version in the online monitoring mode in Unity:

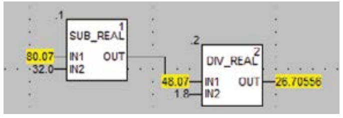
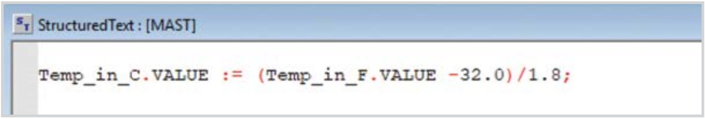


Figure 9

With its powerful formula-processing format, Structured Text makes quick work of a simple math problem like this. In the SCADAPack Logic Editor ST environment, there is no need to do declarations for variables and connection to physical IO or to system tags is done by browsing to the tag you wish to use.

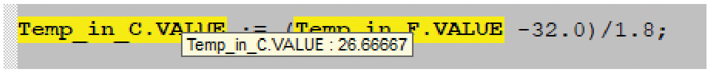
This same application written in ST shows how simple this can be. What you see below (figure 10) is the whole program as written in Unity for SCADAPack. Just one simple line of code that is pretty easy to understand. You type in the correct formula and voilà, magic happens.



```
Temp_in_C.VALUE := (Temp_in_F.VALUE -32.0)/1.8;
```

Figure 10

And in monitor mode:



```
Temp_in_C.VALUE := (Temp_in_F.VALUE -32.0)/1.8;
```

Figure 11

Like the two diagramming languages, Unity allows you to right-click and browse for your variables so typos are easier to spot and remove. In the Structured Text programming equivalent you can even browse for functions and Unity will auto-fill them as needed.

In this task matchup the ST clearly wins, but what about some logic or the use of some function block from the extensive library that installs with Unity? That's where you'll see Function Block Diagrams take the edge.

Below is a self-resetting off timer in LD which is simple enough. When the preset time (PT) is reached, the output (Q) turns off but the little circle at the Q output indicates that it has been negated so the opposite is true. Here is the program in Function Blocks (figure 12):

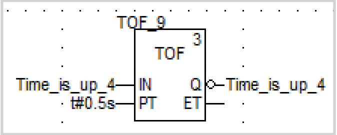


Figure 12

And in the online mode (figure 13):

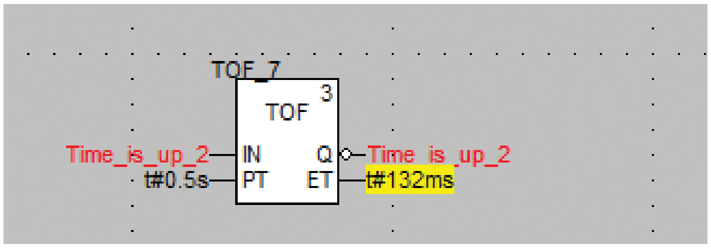


Figure 13

Now with Ladders, as before, except for the EN and ENO connections, there is a strong resemblance between it and Function Block. Here in figure 14 is the same program in Ladders:

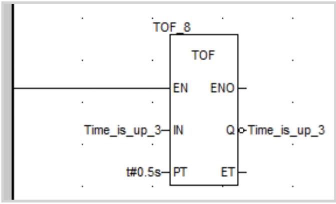


Figure 14

Again, simple enough, figure 15 shows the monitor mode:

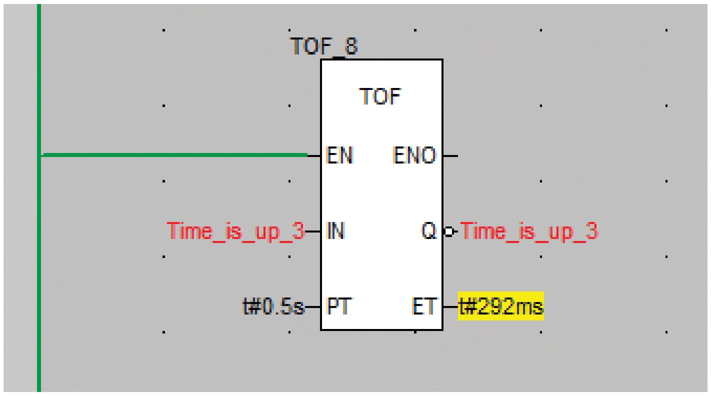
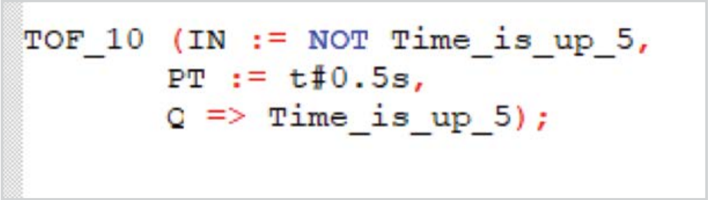


Figure 15

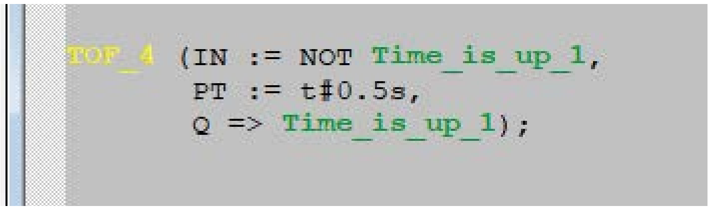
Now, the Structured Text version of the program (figure 16):



```
TOF_10 (IN := NOT Time_is_up_5,
        PT := t#0.5s,
        Q => Time_is_up_5);
```

Figure 16

And in the online mode (figure 17):



```
TOF_4 (IN := NOT Time_is_up_1,
        PT := t#0.5s,
        Q => Time_is_up_1);
```

Figure 17

Not impossible to decipher, but we can definitely start to see some of the complexity that can stack up in this language. This type of line by line code will turn off many non-programmer types whereas the diagramming languages are seen as less daunting. For many systems, the challenge of Structured Text is how to take advantage of the strengths of this program without alienating half the programming team.

Fortunately, there is a solution to the language selection conundrum that can please nearly everyone; Derived Function Blocks. These custom function blocks can be created and used in any of the above three languages so a programmer can make use of the benefits of each without being limited by any one of them. Let's say you need to calculate discharge rates for treated waste water into a stream based on calculated flow rates or into tidal waters based on tide tables. This heavy lifting can be done in ST by programming staff and used by non-programmer types in a more accessible logic environment as a custom Derived Function Block. Another good example would be to calculate and control flow rates through a delivery gates in an irrigation system with a complex algorithm and make use of this in a simple FBD program. This approach has the added benefit that sections of code such as custom functions can be password secured to protect intellectual property or prevent meddling in sensitive code by unauthorized personnel.

In this discussion, you may have noticed that we have have completely overlooked the two other languages in the standard IEC6-1131-3 set (Instruction List and Sequential Function Charts). This was intentional, as these languages do not form the core of a system and function more as a boiled-down or compound version of the other three. In a future article we may expand further on them.

Many control experts today find that the accessibility of Function Blocks with the power of Derived Functions written in Structured Text, Ladders and in Function Blocks provides the best of all worlds. If you would like to see a demonstration of the SCADAPack Remote Connect configuration software which includes the X70, IEC6-1131-3 compliant logic editor and get a free copy of this software, please give your local Sage Designs representative a call at the info below!

SAGE DESIGNS, INC.
SCADA & Security Products

150 Shoreline Hwy., Suite #8A
Mill Valley, CA 94941-3634

Return Service Requested

SAVE A TREE

The Sage Advisor

SCADA, SECURITY & AUTOMATION NEWSLETTER

Calendar of Events

October 21-24, 2019	**CA-NV AWWA Conference,** San Diego CA
November 5-8, 2019	**USCID Conference,** Reno NV
November 19-21, 2019	**RemoteConnect Training Class,** Buena Park CA
December 11-13, 2019	**Colorado River Water Users Conference,** Las Vegas NV
January, 2020	**USBR Mid-Pacific Water Users Conference,** Reno NV
February, 2020	**CA Irrigation Institute Annual Conference,** Sacramento CA
March 10-11, 2020	**TelePACE Training Class,** Corte Madera CA
March 23-26, 2020	**ClearSCADA Level 1 Training Class,** Corte Madera CA
March 31-April 3, 2020	**CWEA Annual Conference,** Reno NV
April 6-9, 2020	**CA-NV AWWA Spring Conference,** Sacramento CA
April 21-23, 2020	**RemoteConnect Training Class,** Corte Madera CA
May 5-6, 2020	**TelePACE Training Class,** Buena Park CA
May 11-14, 2020	**ClearSCADA Training Class,** Buena Park CA

*Download the registration form from our website or call for more information.

Acknowledgements: SCADAPack™, FlowStation™, and ClearSCADA™ are trademarks of Schneider Electric.