HMI HANDBOOK

Practical Guide to HMI Selection, Design and Operation

from AUTOMATIONDIRECT.com

ADC-001-HMI

HMI Handbook

TABLE OF CONTENTS

General HMI

C-more Specific

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look For
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

What is an HMI and What Can It Do for you?

It's well known that human beings are visual creatures, and we would all choose a pretty picture over boring text any day. Just look at the popularity of image-based apps like Instagram or Snapchat and it's obvious that images are king. But on the scientific side, it's believed that our brains can process visual information 60,000 times faster than text, with 90 percent of information transmitted to the brain being visual, and over half of the brain being devoted directly or indirectly to vision. Knowing that, one can conclude that using a visual representation of a machine or automated system would allow for faster recognition and understanding of system variables. And that's where the HMI comes in to play.

What is an HMI?

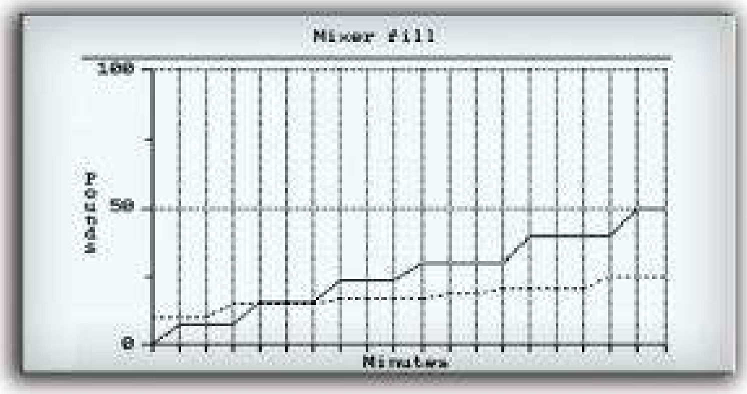
The HMI or Human Machine Interface allows operators to "interface" with the system they oversee. It provides a visual overview of the automated system's status and direct control of its operation. An HMI's graphical screens can be programmed to display important status and control information to the operator. Pictures, icons, sounds, and colors can all be used by HMIs to visually represent different operating conditions. And many HMIs deploy touch screen technology for user interaction with elements displayed on the screen. Now that 'What is an HMI?' has been answered lets talk about what they can do.

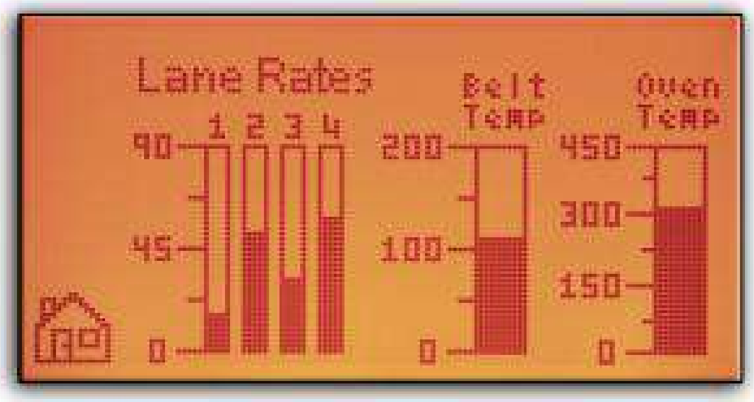
What can an HMI do for you?

Let's say you need to decrease the temperature of a PLC controlled industrial oven in your plant. You could do that by connecting to the PLC with the programming software and making the program changes online. Or you could install a potentiometer to adjust the temperature setpoint blindly without any real-time indication of how the system is reacting. Or you could enter the new value directly by tapping on the HMI screen. By the way, the HMI also displays the current temperature of the oven, a historical plot of temperature fluctuations over time, and will warn you if you set the temperature too low. In this case, the HMI's convenience and functionality wins hands down. And that's just one scenario. There are many situations where HMIs reign supreme over pilot devices. Think about modifications. What would it take in terms of wiring and mounting to install three new pushbuttons on a control panel? Not to mention having to buy them in the first place. With an HMI, it's all in the program. Adding or removing objects, like buttons, on the screen take no time at all and doesn't cost anything extra. In fact, most HMIs have object libraries that consist of numerous buttons, switches, dials, etc. that can be easily inserted into position on the HMI screen – usually with a simple drag and drop. With the many graphical objects in one small HMI you could replace hundreds of mechanical control devices around your plant.

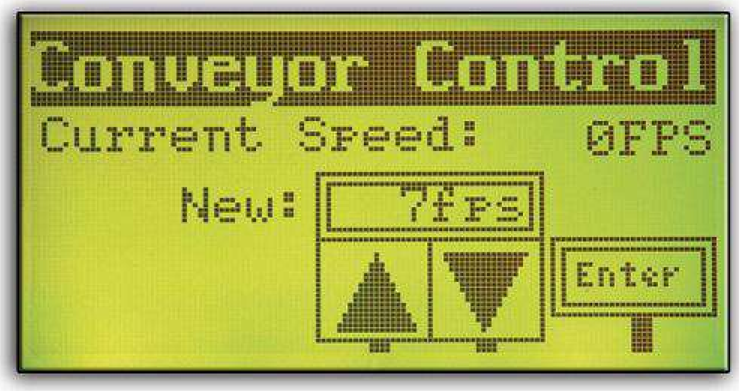
Here are a few more benefits HMIs provide by interfacing seamlessly with a PLC-based control system:

Line/Bar Graphs – Display real-time process values over time with line graphs or in easy-to-read bar graphs.

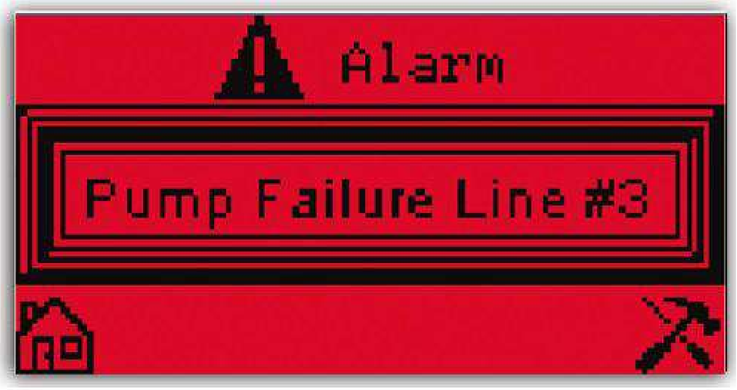




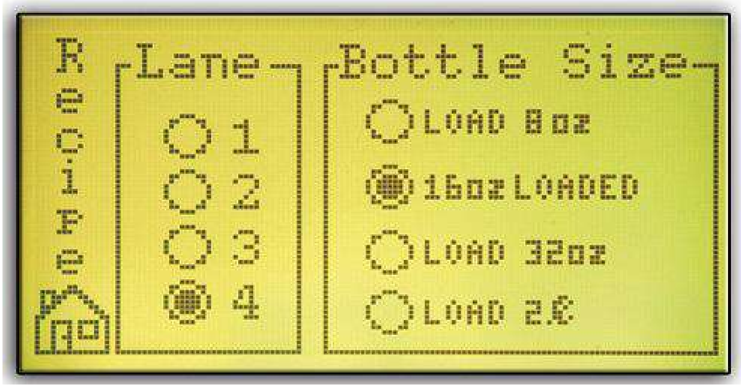
Data Entry – Easily change process variables with a pop-up numeric keypad on the screen or with increment/ decrement arrows.



Alarming – See fault indications with detailed descriptions that aid in troubleshooting.



Recipe Functions – Change numerous operating parameters with one click. Recipe buttons transfer multiple values from PLC source registers to PLC destination registers and/or from the recipe table to PLC destination registers.



Communication – HMIs also provide communication channels for easy upload/download of programs, PLC communication and industrial networking.

Advanced features – The more advanced HMIs can also provide email capability so the HMI can email important data to you, web server functionality so you can log on to the HMI from anywhere in the world, data logging so you can record production data for future reference, and mobile apps for your smart-phone so you can check on production while on the go.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

What does an HMI cost?

HMIs can vary greatly in both performance and price. But for a simple application, one where you are replacing a half dozen indicator lights, micro HMIs can be surprising competitive. Take the C-more Micro HMI for instance. The basic 3-inch model starts at $98* with free software. Now, if you were to purchase 6 panel-mounted signal beacons you might pay as little as $171* (AutomationDirect.com has them starting at $28.50* each). That's a great price but compared to the C-more, you're paying a much higher cost for much less functionality. In fact, you could purchase a CLICK PLC to go along with the C-more Micro and still save money over the beacons:

$171* (beacons) vs $98* (HMI) + $69* (CLICK PLC) = $167* (PLC and HMI)

"But what about wiring and installation?" Just like with the beacons, you would have to mount the HMI to a panel. But instead of wiring numerous signal wires to the indicator lights, wiring an HMI into a control system with a PLC is just a matter of connecting one communications cable (serial, Ethernet, etc.) between the HMI and PLC. The C-more Micro HMI makes installation even easier with applications using AutomationDirect PLCs, like the CLICK, since the HMI can be powered as well through the communications link.

In any project, cost is a primary concern. For simple systems, micro HMIs offer many benefits over mechanical pilot devices including more functionality, time-saving modifications and overall cost. So if you're thinking the budget or the system itself is just too small for an HMI, take a look at the micro HMI market. You might be surprised what you find.

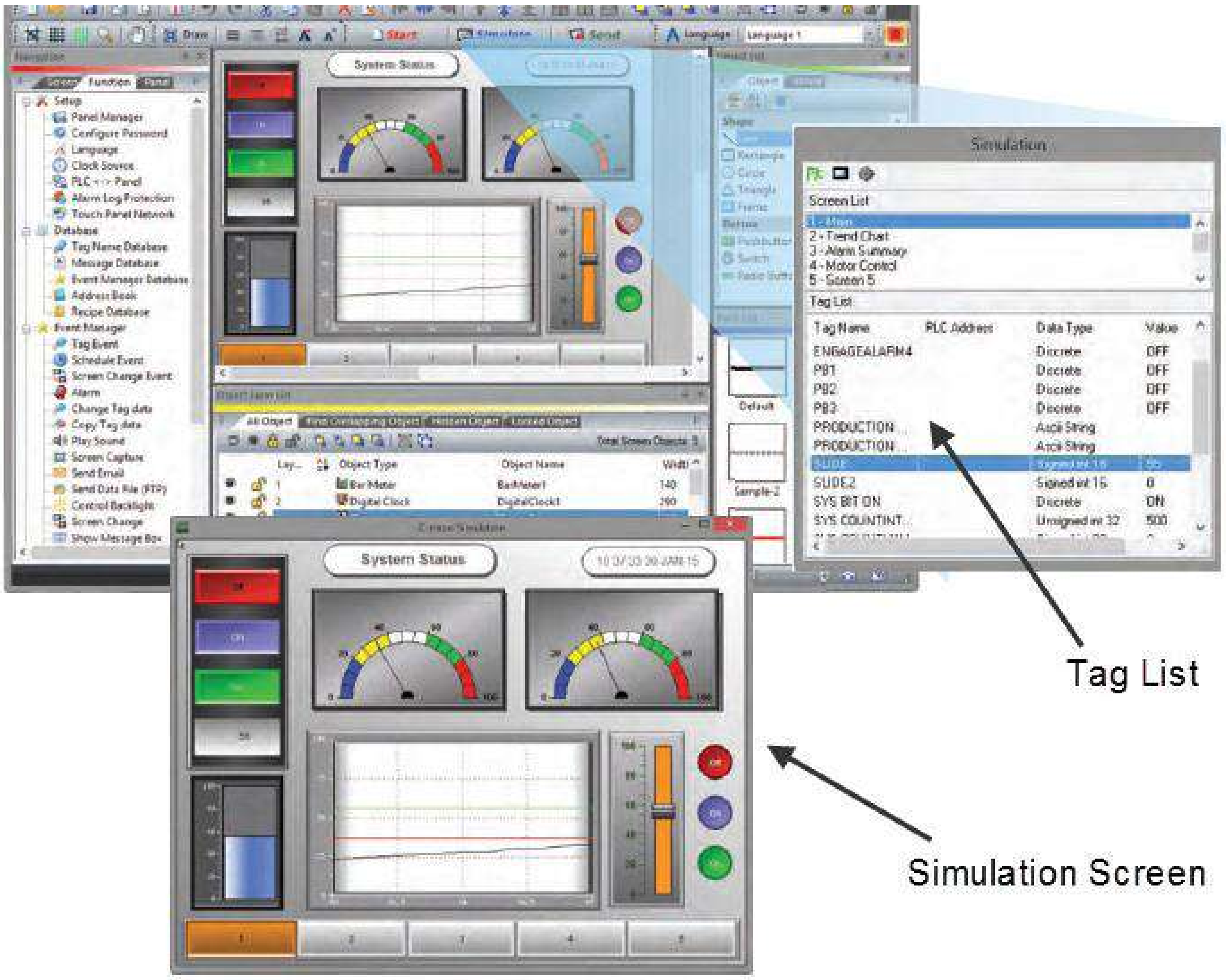
**See our store at www.automationdirect.com for current pricing.*

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Good HMI Software: What Capabilities to Look for

When searching for a touch panel HMI, there are lots of options. And while there are certainly differences in the available hardware, many of the available panels on the market have similar hardware capabilities. But the capabilities of the HMI software, and how easy it is to use, can make a huge difference for your project AND your development time-line.

Let's look at a few features and capabilities that a good HMI software package should have…



Screen List

1 - Main
2 - Trend Chart
3 - Alarm Summary
4 - Motor Control
5 - Screen 5

Tag List

Tag Name	PLC Address	Data Type	Value
ENGAGEALARM4		Discrete	OFF
PB1		Discrete	OFF
PB2		Discrete	OFF
PB3		Discrete	OFF
PRODUCTION ...		Ascii String	
PRODUCTION ...		Ascii String	
SLIDE		Signed int 16	55
SLIDE2		Signed int 16	0
SYS BIT ON		Discrete	ON
SYS COUNTINT...		Unsigned int 32	500

Built-in simulator

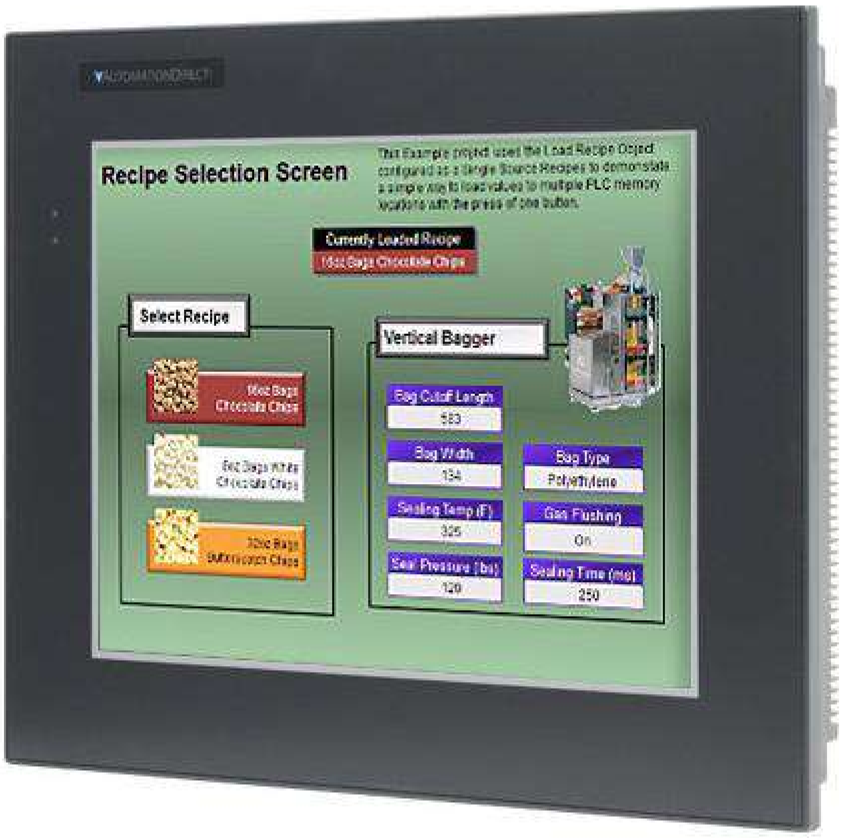
An easy to use simulator will result in a better looking, more effective project AND reduce your development time. You should be able to fully simulate your entire project on your PC (every object, every screen) while it is being developed, without being connected to a touch panel or PLC/PAC. A good simulator should provide a pixel-perfect representation of how your project will appear on your physical HMI panel. A method for editing the relevant tag data in your project should also be provided, with all the relevant tag names listed for each screen that is displayed in the simulator.

You should be able to use your mouse/cursor on the PC screen to simulate a finger touching an actual HMI. And you should see the tag values update accordingly. Conversely, you should be able to edit tag values, and see the HMI screen graphics react in real-time. For example if you want to see how an analog meter moves based on dynamic data from the PLC/PAC, you should be able to simply change that data value and watch your meter move on the simulation screen.

And finally, it should be very easy to move back and forth form the simulation environment to the screen development mode. You shouldn't have to waste time setting up the simulation, or stopping and starting your development system. You'll find yourself using this feature every few minutes while developing your project.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Recipe Functions


AUTOMATIONDIRECT
Recipe Selection Screen
This Example project uses the Load Recipe Object configured as a Single Source Recipes to demonstrate a simple way to load values to multiple PLC memory locations with the press of one button.
Currently Loaded Recipe
16oz Bags Chocolate Chips
Select Recipe
16oz Bags Chocolate Chips
8oz Bags White Chocolate Chips
32oz Bags Butterscotch Chips
Vertical Bagger
Bag Cutoff Length
583
Bag Width
134
Bag Type
Polyethylene
Sealing Temp (F)
325
Gas Flushing
On
Seal Pressure (psi)
120
Sealing Time (ms)
250

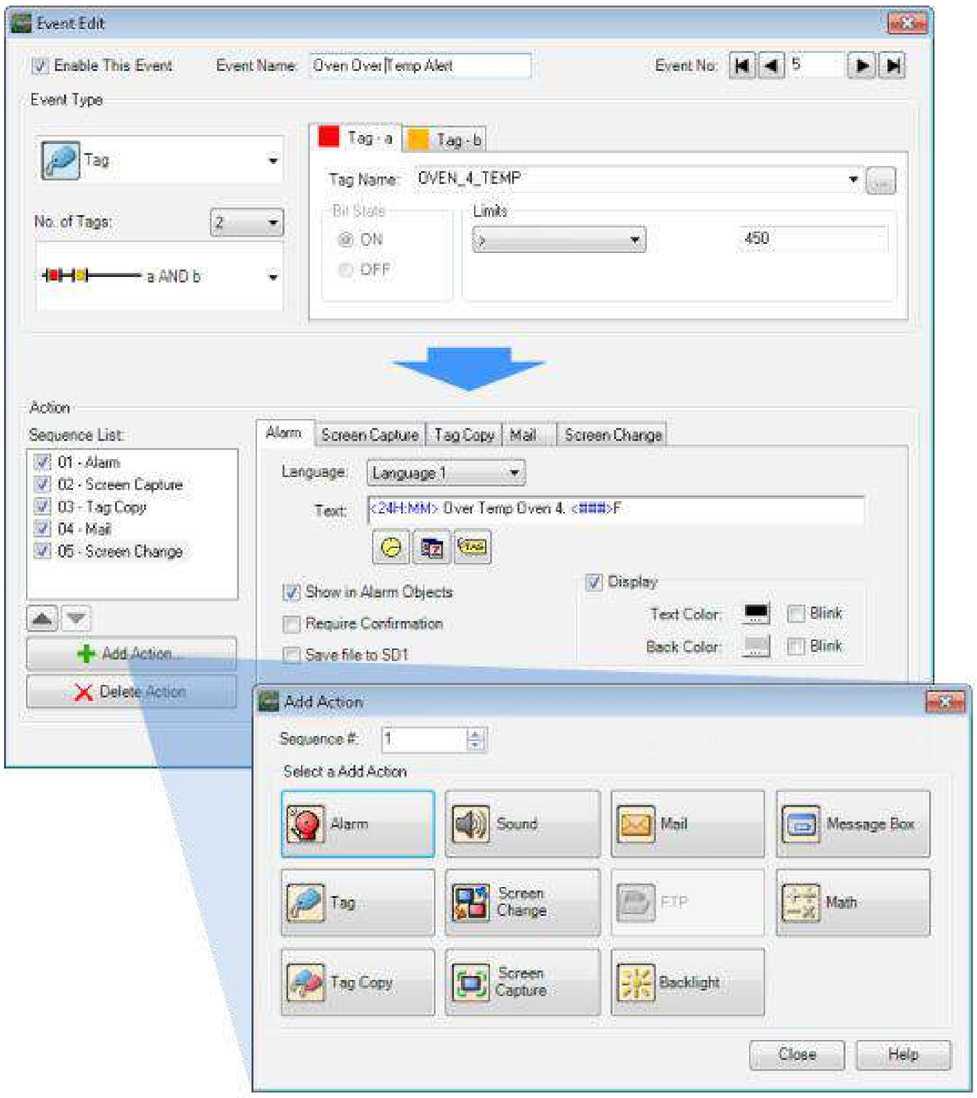
Recipes should make it easy to make large numbers of tag (data value) changes with the push of a single button. Recipes are incredibly useful, not just for the traditional process industries (i.e. mixing of ingredients), but also for machine setup values and other data groups. For example, a recipe can enable a packaging machine to "change over" from jumbo size candy bars to "fun" size in just a few seconds. And the stored parameters reflect the exact setup values as they were the last time you ran that product or operation.

The operator should be able to modify and save recipes on-the-fly while the machine is running (given that the operator has the appropriate level of access). You should also be able to quickly import and export your recipe data to a spreadsheet for further manipulation or archival purposes.

Helpful Event Manager

An Event Manager should provide much more than simple Alarm and Message functions, and should enable an HMI to take action based on random events like changing tag values, or periodic events like day-of-week, time-of-day, or even based on a particular screen change. Besides the traditional alarm display, the action options should include setting of tag values, copying of tag data between tags, changing the screen, capturing a screen, playing a sound file, and even sending e-mail (including embedded tag data, screen captures, or PLC log files).

A good event manager can actually simplify your PLC code, because of all the functionality that the HMI can provide.


Event Edit

☑ Enable This Event   Event Name: Oven Over Temp Alert   Event No: 5

Event Type

Tag

No. of Tags: 2

a AND b

Tag - a | Tag - b

Tag Name: OVEN_4_TEMP

Bit State: ● ON ○ OFF

Limits: > 450

Action

Sequence List:
☑ 01 - Alarm
☑ 02 - Screen Capture
☑ 03 - Tag Copy
☑ 04 - Mail
☑ 05 - Screen Change

Alarm | Screen Capture | Tag Copy | Mail | Screen Change

Language: Language 1

Text: <24H:MM> Over Temp Oven 4. <###>F

☑ Show in Alarm Objects
☐ Require Confirmation
☐ Save file to SD1

☑ Display
Text Color: ☐ Blink
Back Color: ☐ Blink

Add Action...
Delete Action

Add Action

Sequence #: 1

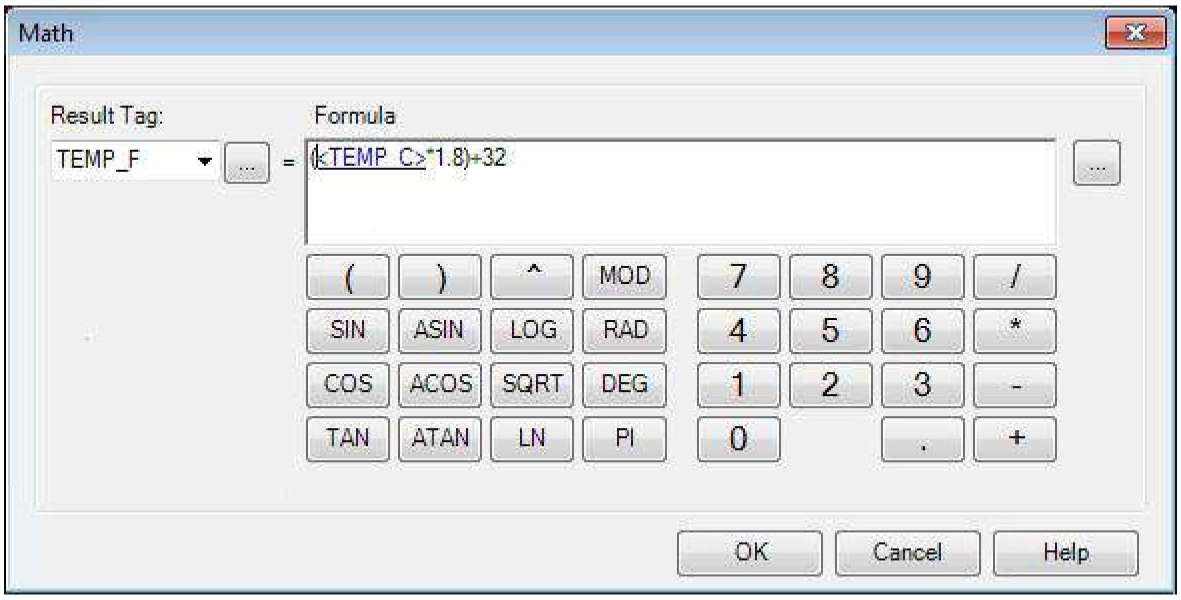
Select a Add Action

Actions should be "triggerable" by various events, such as:

- Any combination of the following with one OR MORE tag values:
    - Tag values turning ON or OFF (bit tags)
    - Tag values equal-to, above or below a specified value
    - Tag values inside or outside a specified range
- Day and/or Time triggers
    - Every day or specific days of the week
    - Specific times of the day
    - Repeating every X hours and/or Y minutes
- Based on a Screen Change
    - Use to log visits to a particular screen
    - Also handy when a calculation or some other action needs to occur before a screen is displayed

Supported actions should include:

- Triggering Alarms
- Writing Tag values
- Copying one tag value to another
- Playing a sound
- Changing the currently displayed screen
- Capturing an image of the current screen
- Sending an e-mail, including embedded:
    - Tag Data
    - Date and Time
    - Screen Captures (as attachments)
- Initiating FTP (File Transfer Protocol) uploads to a server
- Popping-up message boxes (with tag data, date/time, etc.)
- Performing math calculations. . .



Result Tag: TEMP_F

Formula
= (<TEMP_C>*1.8)+32

It is very useful to be able to perform mathematical calculations when a particular event occurs. You should be able to create custom formulas that include any combinations of constants and live tag values, and to store the result into a user-defined Tag, or display it on screen, or send it via e-mail, etc.

Jump to Chapter

Remote Access

HMI software should allow remote access to your HMI in several ways:

Access and control your HMI from any connected PC (full control, just like you are standing in front of the panel). And the interface should be browser based so that you don't have to purchase any additional software for the PC.

Authorized PC users should have the ability to remotely:

- Monitor and control screen operations of the panel as if touching the panel itself
- Test and troubleshoot the HMI project
- View, zoom, print and save screen captures of active screens

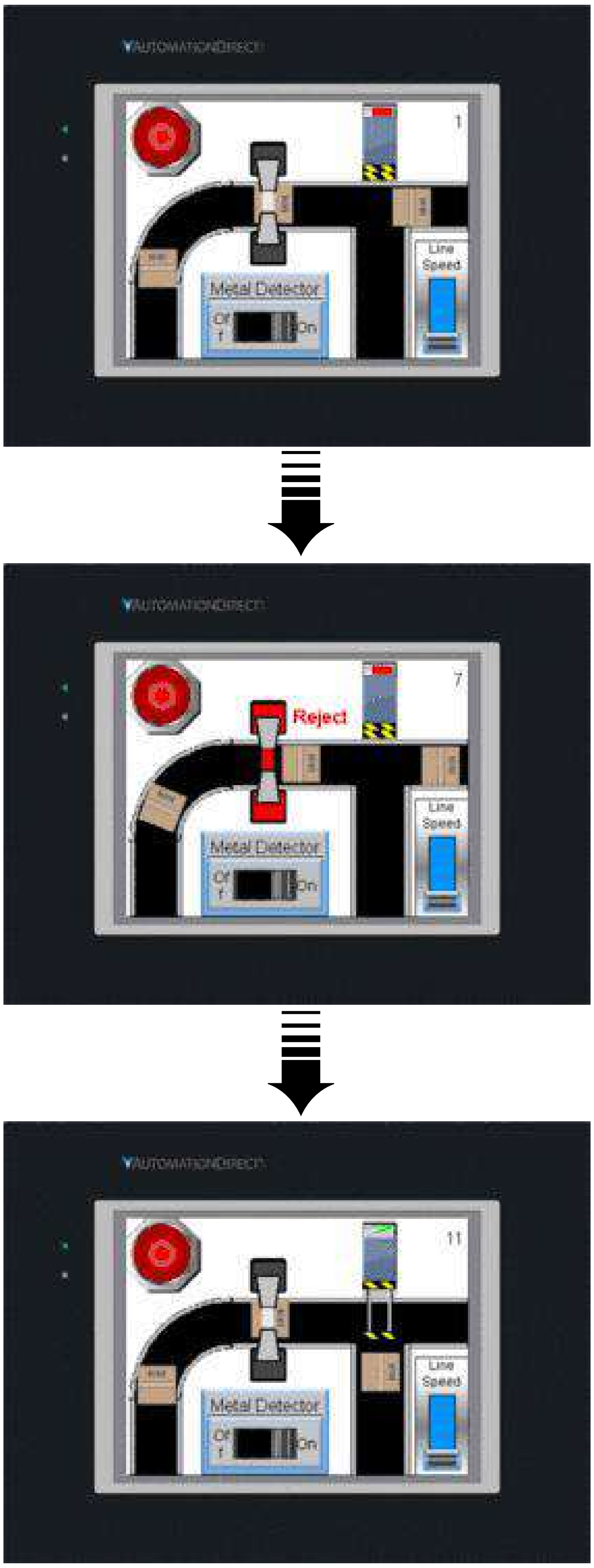
But it shouldn't stop with PC-only remote access…

Access capabilities from your iPhone, iPad, or Android mobile device should include:

- Monitoring and controlling screen operations of the panel as if touching the panel itself
- Users should be able to save screen captures to review, email or print
- Should allow Zoom, allowing the user to zoom in on specific objects and then save screen captures if needed
- Should allow Multi-level Login Security
- Should allow simultaneous access for multiple remote users
- Multi-level Access Control should allow various levels of access: Full control access, View only access, View and Screen change only access
- Activation or Notification Tags should provide an option for each account to have associated user configurable tags that allow the Panel project or PLC project to indicate that a remote user is connected or to enable or disable the remote access feature. These tags should be able to activate alarms, events or notifications to alert local operators that a remote user is connected. The Disable/Enable tags should even allow local operators to enable or disable the remote access feature for security or safety reasons.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Animation



Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

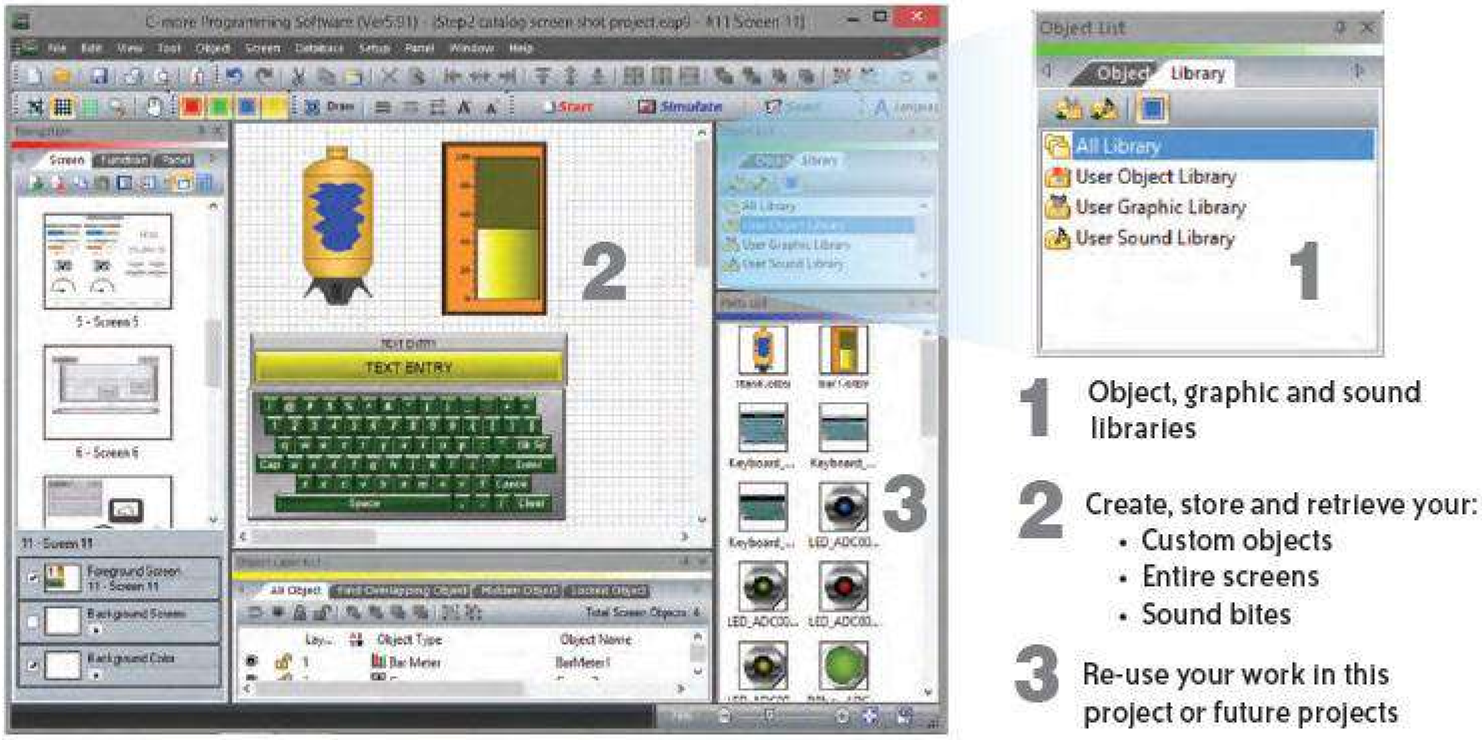
Good HMI software should make animation easy. Animations of machine motion and factory processes can enhance comprehension, and give operators a visual representation of the desired motion of products, and machinery on screen. In fault conditions, operators can quickly see the relationship of product or machine position to sensors, or fault zones. Animation can be used to teach operators the correct relationships between mechanical elements or product positions.

Animation capabilities should include movement, rotation, sizing, visibility, and even the use of sequentially displayed images to simulate motion.

Easy to Use Development Tools

Object, Graphics and Screen Libraries

Once you've created an object, a group of objects, or a screen – you should be able to give it a name and save it within the HMI software for fast and easy reuse later. You should be able to store objects, graphics, sounds, and even entire screens in this manner. You should have the option to save objects/screens with or without the PLC tag info, and to easily share objects/screens from your library with other developers that you may work with. Reusing common objects and screens not only saves time but also creates a common look or theme that your operators will benefit from.



1 Object, graphic and sound libraries

2 Create, store and retrieve your:
- Custom objects
- Entire screens
- Sound bites

3 Re-use your work in this project or future projects

Object Layer List

An Object Layer List should be incorporated. This list provides a rundown of all the objects on the active screen and allows you to quickly select an object from the list to edit. With this list you can expand or collapse grouped objects and access individual objects in a group without having to separate the group or move objects on the screen.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

And More…

Good HMI software should allow you to transfer a project to an HMI without having to take the programming PC to the panel using a flash media card or a USB memory stick. This is ultra-convenient, whether the panel is installed in your local factory, across town, or on the other side of the planet. Project transfer using Ethernet or USB should also be supported.

Good HMI software should offer database tools such as search (with filtering), and the ability to import and export all the various databases for large projects. Separate HMI databases should include: Tag-names, Messages, Recipe values, Address Book, Event Manager, Multi-Language Text Strings, Object and Screen Libraries, and even Screen Images. These features alone can save countless hours of development time.

Good HMI software should allow wide flexibility in the use of typefaces, fonts, and text size and should allow all text to be displayed in multiple languages. Language selection should also be selectable with the touch of a button.

Good HMI software should have a flexible, dock-able windows layout and offer drag and drop functionality for all screen objects.

And finally, a good HMI software package should preserve both tag data, with easy to use data logging capabilities, and project data, with auto save and backup functionality, lapse grouped objects and access individual objects in a group without having to separate the group or move objects on the screen.

Weigh your options…

When it comes to HMI products, there are plenty of options out there. What's listed here are some of the features that a good, quality HMI software package should employ. HMI development can be simple or extremely complex but either way, the HMI software should provide the user with the functionality and easy-to-use tools needed.

If you want more information on HMI features you can check out the C-more HMI feature section of our website at https://www.automationdirect.com/c-more/software/explore. There you'll find many of the features mentioned here in greater detail, plus a few extras. If you are in the market for HMI products, check out the full C-more EA9 series at https://www.automationdirect.com/c-more/index. There you can see informative product videos, read customer reviews and see everything C-more has to offer.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Pick and Place Your HMI

There are many ways to irritate an operator working with automated equipment and a bad human-machine interface (HMI) implementation is right up there with the worst of them. Following these guidelines can help you pick the right HMI for your application, keep you on the good side of your operators and engineers, and also help your machines and processes run more smoothly and safely.

Many types of HMIs are available in today's market from simple operator interface terminals (OITs) with two-line displays to more capable embedded HMI graphics terminals to powerful PC-based HMIs. Here's how to select the best one for your application.

Pick the Right HMI

When selecting an HMI for a small machine, an OIT is typically acceptable—and certainly simple to configure, easy to use, and low in cost. An OIT will connect directly to the machine controller and provide suitable local operator interface functionality. It is also a good choice when adding additional operator displays to large assembly lines to give operators a better view of equipment status and line operation.

With larger machines or complex manufacturing lines, more extensive capabilities may be required, and this is where PC-based HMIs can be the best choice. With these types of HMIs, the PC can be industrially hardened for environmental reasons or office-grade. PCs today are available with multiple processors, large hard drives, lots of memory, and include a wide range of built-in connectivity options. High-end functions such as a historian and the ability to simultaneously supervise multiple production processes can quickly justify the extra cost of this option.

Not Too Big, Not Too Small, Often Just Right

The obvious choice for mid-size equipment and processes is an embedded HMI. However, in many cases, this middle-of-the-road HMI can also fit the bill for applications small to large. Even smaller machines can benefit from connection to the cloud, and these 6- to 15-inch embedded HMIs work well in a variety of configurations whether standalone or networked.

Embedded HMIs can go it alone, but they also play well with others. In a dedicated, standalone setup, an embedded HMI connected to small or mid-sized automated equipment is a cost-effective and simple solution. Simply ensure the correct protocol is available to talk to the machine controller, hardwire it to the controller with its included cables, and you're ready to configure the software. Unlike OITs, these embedded HMIs have many features including the capability to interface with mobile devices such as smartphones and tablets. If you're working in a networked setup, there are two main network architecture options.

HMI Network Architecture Options

There are two types of networked HMI solutions, one with multiple HMIs connected to one controller, and the other with one HMI connected to multiple controllers. Some applications work best with both options employed simultaneously as multiple HMIs are connected to multiple controllers.

OITs and embedded HMIs fit well into the first scenario, where multiple HMIs are connected to one controller on a single large item of equipment like a printing press, or an assembly line with multiple manual and automated work cells. In either configuration, this networked solution will support many HMIs. If one HMI fails, other HMIs can provide redundant operator interface to minimize downtime. The fall-back HMI may be in a less than optimal location, but it can provide needed functionality until repairs are made.

In the second scenario, where one HMI is connected to multiple controllers, a PC-based HMI is typically the focal point as it can connect to many controllers simultaneously, even using different protocols if necessary. A PC-based HMI can contain many screens, limited only by the designer's imagination and available memory, so each controller can have its own set of displays. With this option, OITs and/or embedded HMIs are often used to provide local operator interface, with the PC providing an overall view of operation.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Location, Location, Location

A big consideration is where and how to position the HMIs. It is very important to consider ergonomic and normal operator position. If the operator has a kink in his neck from looking at an awkwardly positioned HMI screen, he or she may start to ignore the screens and follow gut instinct. In general, each HMI should be positioned so the operator doesn't need to look up much, and down just a little.

Speaking of location, it's important to consider where the operator is going to spend most of his or her time. How far will the operator have to walk to view the HMI? HMIs are often mounted in the main control enclosure, often located at the front end of the production line. But if there are locations where there tend to be line disruptions or stoppages, how much time would be saved if the HMI were mounted close to where these problems typically occur?

In order to maximize efficiency, it is good practice to mount the main HMI in a centralized and easily accessible location to keep the operator's steps to a minimum. Fewer steps combined with additional OITs and/or embedded HMIs at strategic problem points can quickly pay for themselves in terms of improved operations and safety.

Another location consideration is how far the HMI will be from the actual equipment area of interest. Being 20 feet away from the tooling station you are trying to operate manually from the HMI is never an enjoyable experience. Consider mounting the HMI on a swing arm if the machine footprint allows, as this will make the viewing experience up close and personal.

There are many considerations when designing and integrating a control system's OITs and HMIs, with some of the leading factors discussed here. Once the type of HMI is selected and communication features and functionality are confirmed, consider how the operator and manufacturing engineer will use the OITs and HMIs. Make sure they are readily accessible, and remember that with OITs and HMIs, more is often better in terms of improved operations and safety.

16 Tips for an Effective HMI

Do you have an HMI project coming up? Here are some helpful tips and guidelines for you to remember during your planning and implementation.

1. Story Boards are a Good Start

A good starting point for HMI design is a text-based outline documenting each screen's content. With the operator's point-of-view and ease-of-use in mind, detail the main screen, equipment status screens, set point or recipe screens, manual functions, message displays and fault displays. These text outlines can then be converted to a story board for each screen.

The story board should highlight dynamic graphics such as status indicators. It should also include repeated graphics, such as titles, at the top of the screen and navigation buttons at the edges of the screen. The screen buttons, indicators and numeric displays should be aligned and grouped as appropriate. For example, don't mix screen selector buttons with start/stop pushbuttons.

2. Use Color… Judiciously

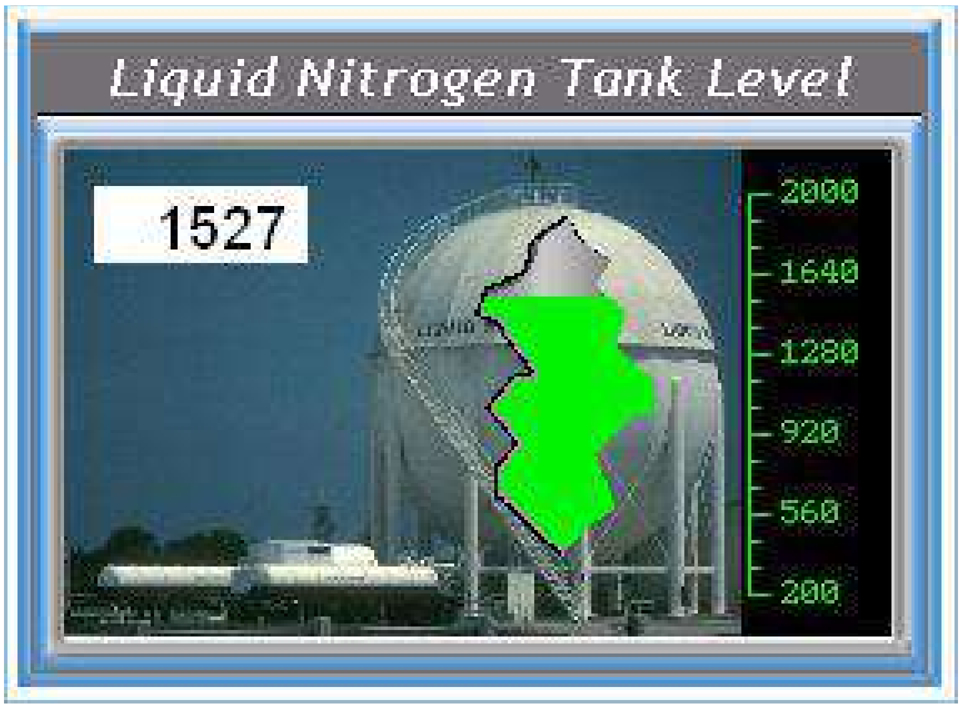
Many of the HMI guidebooks recommend limiting the use of color and using low contrast gray backgrounds. This makes for a much less cluttered screen. Consider a light gray screen background where a typical indicator would be dark gray in the OFF state and white in the ON state. This is easy on the eyes and makes sense as a light bulb turns white when on. Try to use bright, saturated colors only to indicate abnormal conditions.

3. Display Data Completely

Data display is a key component of HMI design, and different data will require different display types. A number on a screen may accurately display speed, but the engineering units may be in question and the acceptable range unknown. This can be solved by adding the units, inches per second for example, and the maximum and minimum values in a table fashion.

4. Add Graphics

The phrase “A picture is worth 1,000 words” couldn’t be truer for HMI design. Operators really do grasp the meaning of an image faster than they can read the text on a boring gray button. And if there is a language barrier, the images become even more important. C-more HMIs allow overlapping objects, so consider allowing a small graphic to overlap a button. There is also a Bitmap Button Object that allows you to use a built in graphic or your own user graphics for the ‘ON’ and ‘OFF’ states. Keep in mind that excessively large graphics and animation effects can slow the performance of a particular screen, so use them appropriately or isolate them on alternate screens to ensure fast response on the primary screens.



5. Display Appropriate Images

An image of the machine being controlled, or other real-world images can be incredibly useful to help operators understand the location of problems like blocked actuators or sensors. Consider how a modern office copier can lead a complete novice to the location of a paper jam in seconds using clear images of the machine itself, locational pointers and even animated images of the flaps and hatches that must be accessed to clear the jam. But resist the temptation to model the entire P&ID piping diagram of the plant (or other process) – ask yourself if it’s just a pretty picture with numbers sprinkled over it, or an intuitive display of relevant information.

6. Animation for the Sake of Animating

With the requirements defined, use caution when developing the graphics, as this is not the opportunity to express your inner Picasso. Some of the more artistically talented HMI designers enjoy using all the colors and graphic animations available in the HMI software. Showing the pumps and fans spinning, valves opening and closing, and fluid moving in the pipes is cool—but may be excessive and can distract the operator.

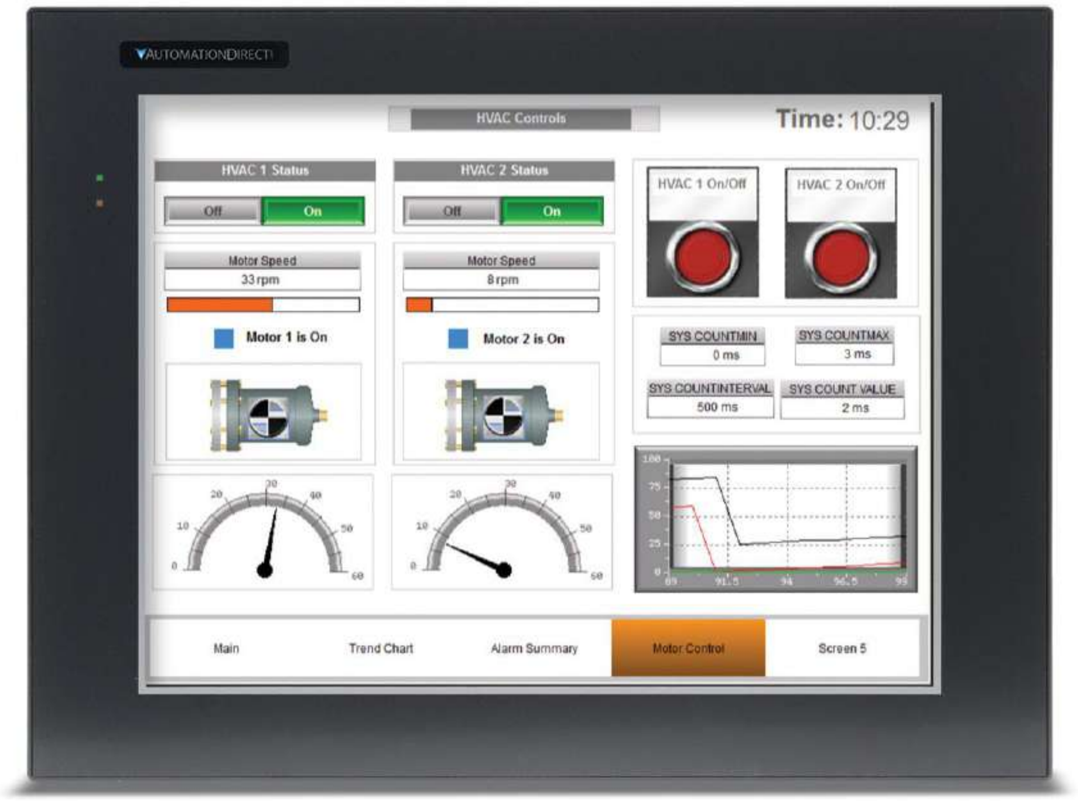
For example, using animation to show the position of a box on a conveyor is an efficient and quick indication of conveyor status. However, showing a pump motor rotating could be a distraction if the intent of the screen is to show high level fault indications. Only animate if it makes the operator more efficient.

“There’s a frustrated artist residing in many of us, but HMI screen design isn’t the time or place to express these urges.”

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

7. Keep Important Controls (Stop, Start, Setpoints, Navigation, etc.) Available At All Times

Reserve a portion of the screen for these items; perhaps a band at the top or on the side for these controls, and make sure this area is completely consistent everywhere it is displayed. C-more offers a "background screen" feature that allows designers to create, edit, and maintain such an area in a single location within your project layout, and then display it across multiple screens as desired.



Time: 10:29

Motor Speed 33 rpm

Motor Speed 8 rpm

SYS COUNTMIN 0 ms

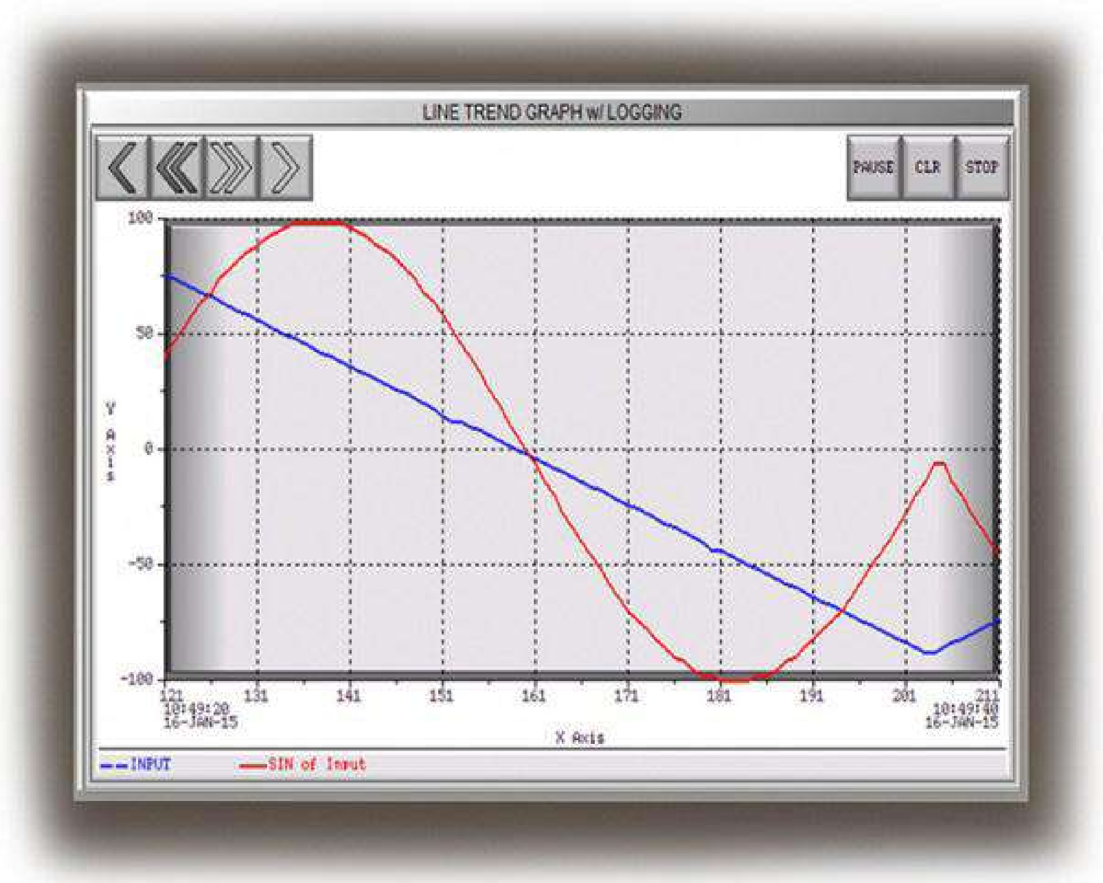
SYS COUNTMAX 3 ms

SYS COUNTINTERVAL 500 ms

SYS COUNT VALUE 2 ms

8. A Trend Graph Can be Useful

A line trend graph can display past and present data and is a good indicator of future values as an operator can quickly see the values trending towards an upper or lower limit. In this case, displaying the actual numeric value is not necessary as the data is displayed graphically.



Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

9. Offer Situational Awareness

Make sure relevant data is displayed clearly, so that operators can grasp the current state of the machine or process at a glance (or as quickly as is practical). A good display answers both questions: "Where is the data/process now?" and "How does that compare to optimal conditions?" For discrete manufacturing, displaying data with respect to the "present state" of the machine may be enough, but for more complex machines or processes the screen design and layout should strive to help an operator predict the future status of a machine. Is it enough to know the current temperature of the oven, or would a trend graph allow the operator to foresee/prevent a pending disaster?

There are many layout and design factors that can detract from situational awareness. Data overload and focus on the wrong information are two of the most common. If the situational awareness of the operator is critical to the success of a particular HMI project – there is a wealth of additional information available. One particularly good resource is: "Designing for Situation Awareness: An Approach to User-Centered Design" by Mica Endsley, Betty Bolté, and Debra Jones.

10. Keep All Info and Controls within 2 to 3 Clicks of the "Home" or Main Screen

Consider the work-flow for the application. Which tasks are performed most often, and which are seldom used? Try to optimize the layout to be as efficient as possible. Keep screen menus and/or screen change actions as consistent as possible throughout the application and have a clear and consistent "back path" so that even the least experienced operator cannot "get lost."

11. Watch an Actual Machine Operator to Learn Their Interactions with the HMI

Using your storyboard, "play act" the HMI's operation with an experienced operator and deploy a prototype of the HMI before final commissioning if possible and watch how the operator uses the interface.

Look for awkward situations. Is there additional automation that can relieve the operator of needless button presses, or otherwise make them more efficient? Ask the operator for advice and opinion, but filter it, as the operator may not see the big picture. Realize, at the same time, that your engineering knowledge of the machine may cloud your perspective on what is actually required to operate the machine efficiently. Remember that a professional driver can operate a racecar at the edge of its performance capabilities without understanding every detail of the engine, chassis, and body design.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

12. Give Feedback and Lead the Way

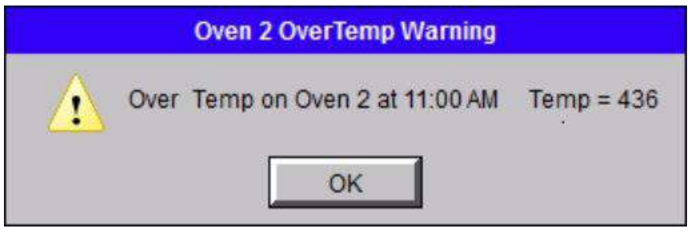
Give the operator visual (and audible) clues that buttons have been pressed, or that certain steps have already been performed. Affirmative feedback can build confidence, and satisfaction with the system. Use color and/or animation to lead operators step-by-step through complicated processes if necessary. But keep things as simple as possible and avoid unnecessary complexity – long lists of operator tasks may indicate "opportunities" for additional automation or other workflow issues. Two good questions to ask, from the operator's perspective: "Where were you expecting to find that?" and "What were you expecting to happen?"

13. Beware of "Pop-up" Alarms or Dialog Boxes

Make sure that "pop-ups" are used sparingly, if at all. Nothing is worse than a cascade of error messages – all of which have to be acknowledged individually – before the operator can get to the screen where the condition can actually be resolved. Other techniques that can be used to draw the operator's attention are simply changing the background color of an object, or section of the screen, or blinking objects (although this too, should be used sparingly). Overuse of blinking or flashing effects can habituate the operator to the effect, and thereby diminish its effectiveness as an alert, or it may even distract the operator from a more important issue elsewhere on the screen.

14. Date and Time Stamp Alarm Logging

Don't just track the alarms – but also when they occurred. Many seemingly unrelated, recurring issues, may actually correlate to shift changes, coffee breaks, or even the startup/shutdown of nearby equipment or other periodic plant operations. Use alarm confirmation requirements to see how quickly operators are reacting to certain conditions and consider refining the HMI layout/design for faster response, if desired.


Oven 2 OverTemp Warning

Over Temp on Oven 2 at 11:00 AM Temp = 436

15. Use Password Protection… Appropriately

Carefully consider what screens or objects should be available to all personnel, and which might be restricted to maintenance, engineering or management staff. Consider safety and the complexity of the items in questions. Realize that an overly restrictive policy can hurt productivity, or worse: result in the sharing of passwords with unintended personnel simply for the sake of "up-time".

16. Create a Style Guide

Create a set of common styles to ensure consistency across all HMIs in a particular factory or plant. Consistent use of similar indicators, graphics, trend objects, etc. can increase operator familiarity and improve comprehension. C-more offers both an object library and a screen library, where you can store (and share) proven elements – or entire screens – for easy reuse across multiple projects.

One additional resource that may be useful, especially for process industries, is "The High- Performance HMI Handbook" by Bill Hollifield, Dana Oliver, Ian Nimmo, & Eddie Habibi. The book reveals many poor yet common HMI practices, provides the justification for change, and shows in great detail the best way to design and implement HMIs. It contains over 90 color illustrations, and many real-world examples.

There are many ways to develop HMI screens for machine automation and related applications, but successful implementation requires discipline in design. Guidelines, standards and handbooks covering HMI best practices and design include those published by ISA, ASM, ISO, NUREG and others. All of these standards discuss a wide range of design, build, operation and maintenance methods for effective HMIs. Many also discuss safety, quality, reliability and efficient control of the equipment or process under normal and abnormal conditions. These standards can be used as a basis to create internal company HMI design guidelines, which in turn can be used to create consistent and effective HMI screens from one machine or project to the next.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Remote Monitoring with Embedded HMIs

If your company has remote sites that need to be monitored, and if you're looking for a way to do this without making periodic rounds to check each site, then remote monitoring may be the answer. And if you're looking for a cost effective yet powerful way to implement remote monitoring, embedded HMI is a solution worth considering.

PC-based HMIs work well to provide local operator interface and remote monitoring at many pumping stations and other remote locations, but embedded HMIs can provide many of the same benefits while saving time and money.

Why Monitor Remotely?

Many remote locations have been running unattended for years without remote monitoring, but companies are now implementing off-site access because it allows a single user to monitor and maintain many remote sites. Not only can a single user monitor multiple sites remotely and simultaneously, he or she can solve problems without going to the site by accessing the HMI directly, and the control system through the HMI.

And, this monitoring can be automated by having emails and text messages pushed to personnel, alerting them to immediately respond to any pressing issues. Once alerted, remote personnel can access the HMI using any web browser.

To PC or Not to PC

Everyone knows what a PC is, and a PC-based HMI is simply packaged software running on a PC. Alternately, a PC-based HMI can be software written by the end user, but this approach is less common.

Embedded HMI is a less familiar term and requires more explanation. Control and monitoring software running on an embedded operating system (OS) constitute an embedded HMI, like AutomationDirect's C-more. An embedded OS requires much less processing power and memory than a PC-based HMI because it performs a single function, in this case hosting the HMI software.

This makes embedded HMI less expensive than PC-based HMI, and also provides greater OS and application longevity, along with these other advantages:

- More compact
- Simpler maintenance
- Easier installation
- Configuration software less complex
- More suited for harsh environments

Going Mobile

Remote support personnel can access embedded HMIs on any platform capable of running a web browser including PCs, smartphones and tablets. This is because most embedded HMIs, including those from AutomationDirect, have built-in web servers.

To establish an embedded HMI as a web server and remotely access it via any web browser, simply follow these steps:

1. Establish an HMI Internet connection
2. Configure the HMI's IP address
3. For each remote user, setup login security, password and access level at the HMI

Each remote user can now access the embedded HMI through any web browser using the login credentials configured at the HMI

PC-based HMIs are needed for some complex applications, but most remote sites can be effectively monitored with an embedded HMI—saving money and space while simplifying installation, setup and maintenance.

While web server capability is a definite benefit of many modern-day HMIs, some HMIs, like the C-more, take it one step further by offering mobile apps. The C-more HMI mobile app is available to purchase ($4.99) for Android and Apple devices and provides a convenient method to stay connected to your remote systems.

See more about the C-more Remote HMI App in this video

Learn More...Watch The Video!

C-more Remote HMI App - Productive, Practical and Priced Right

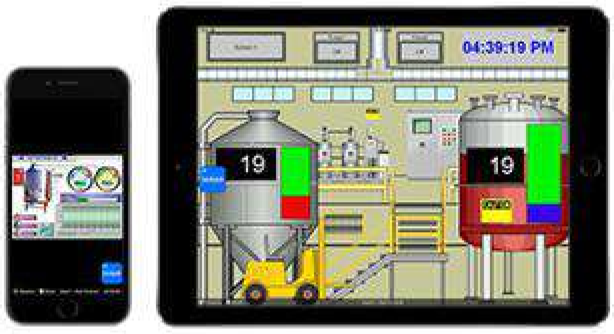
http://n2adc.com/6qkdl

But Why Not Just Use a Browser?

While five dollars is certainly cheap, why would users need an app when smart phone browser-based access is free?

There are certain differences between the methods of access. Remote browser access provides a snapshot of the HMI screens, which is important when monitoring the plant from a computer or other Internet-connected device. However, the C-more iPhone app offers a live view of the screens. Users have near real-time views of monitored variable changes.

The most important difference between the two is in terms of control. The browser access provides read-only access, while the iPhone app enables users to actually control the plant remotely.

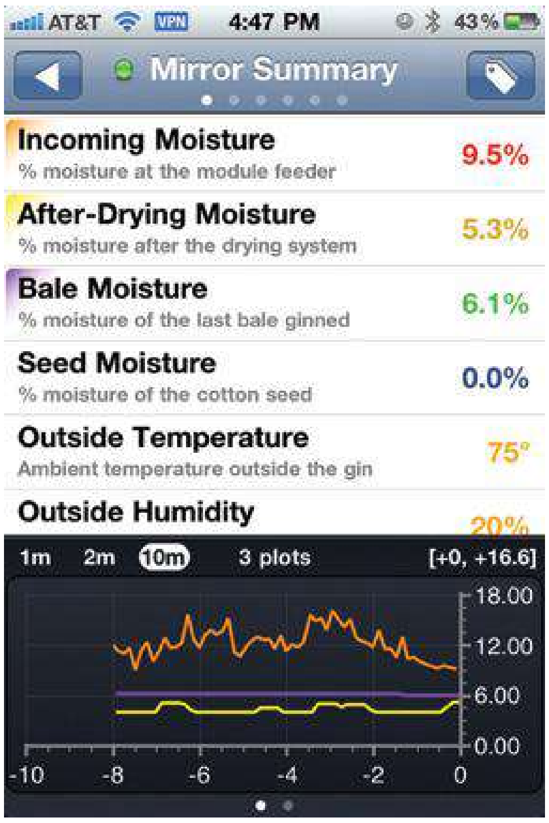


Security

Naturally the question of security is at the forefront of users' minds. We've all heard stories about cell phones and Internet sites being hacked. It can be relatively easy for a hacker to gain access to almost anyone's connection in a public place, such as an airport equipped with Wi-Fi. Therefore, it's important that remote access be carefully engineered using qualified resources, as well as a proper evaluation to ensure there's a good cost/benefit ratio.

There are methods for making a remote connection more secure, including:

- Setting up login credentials and access levels within the HMI.
- Using a VPN connection to a Local Area Network (LAN).
- Cloud computing security can be improved by giving rights to the application running on the device. This allows access to data only when the user has an approved device and a proper login and password.


AT&T VPN 4:47 PM 43%

Mirror Summary

Incoming Moisture 9.5%
% moisture at the module feeder

After-Drying Moisture 5.3%
% moisture after the drying system

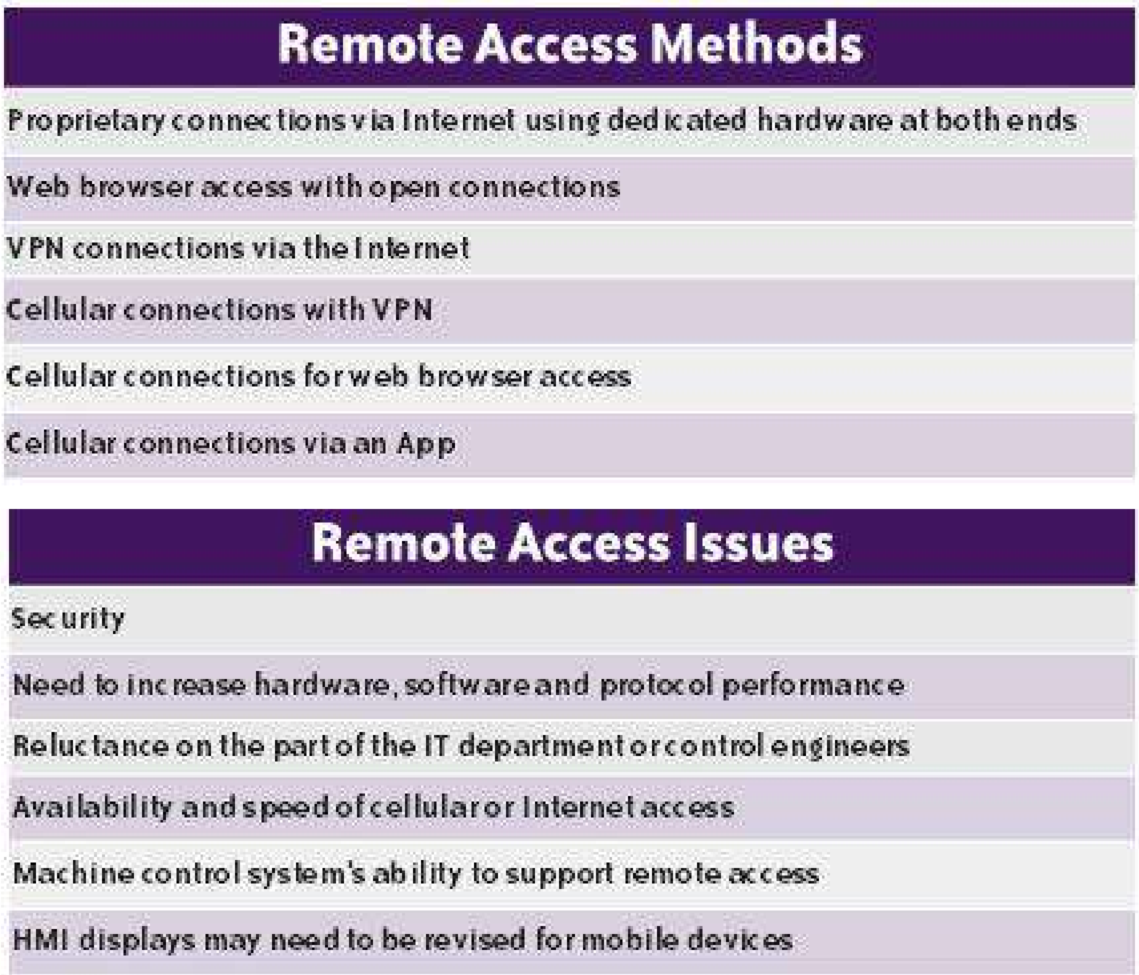
Bale Moisture 6.1%
% moisture of the last bale ginned

Seed Moisture 0.0%
% moisture of the cotton seed

Outside Temperature 75°
Ambient temperature outside the gin

Outside Humidity 20%

Chapter 5



Remote Access Methods
Proprietary connections via Internet using dedicated hardware at both ends
Web browser access with open connections
VPN connections via the Internet
Cellular connections with VPN
Cellular connections for web browser access
Cellular connections via an App

Remote Access Issues
Security
Need to increase hardware, software and protocol performance
Reluctance on the part of the IT department or control engineers
Availability and speed of cellular or Internet access
Machine control system's ability to support remote access
HMI displays may need to be revised for mobile devices

Using C-more Remote HMI App with secure VPN connection

Learn More...Watch The Video!

C-more Remote HMI: Connect Mobile App to Control your C-more HMI

C-more Remote HMI: Connect Mobile App to Contr...
Watch later
Share
C-more® HMI
Remote HMI Mobile Access APP Connecting
C-more Remote HMI App For only $4.99 you can have remote access and control on the go! Available now
App Store
Google play
amazon

http://n2adc.com/pvbwz

Many times, the type of security required depends on the type of user.

- Smaller companies typically don't have dedicated IT security personnel. Their security may consist of a firewall, an Internet router that uses Packet Address Translation (PAT) as well as an operator opening the connection from the machine or robot.
- Large corporations typically have a dedicated IT security staff as well as advanced network architecture. When working with these companies, a path to the machine or robot is necessary. This involves numerous security policies and considerations, which can be cumbersome.

In addition to IT department considerations, the technology and network settings might be a bit daunting to some plant managers and control engineers. Therefore, good security measures must fit the type of organization as well as the people who will be interacting with it (i.e., the IT department, plant managers, control engineers and operators).

While remote monitoring and access via embedded HMIs and/or cellular technology may be simple to provide, there are some security risks that need to be considered. However, as businesses become more global and communications more important, having the ability to offer remote access may be the deciding factor in who is awarded a contract.

Layered Security Goes Beyond Hardware

Industry is certainly starting to take note of the IIoT, which raises cybersecurity issues because it requires connecting field devices and other automation system components to the outside world, often through the internet. Connectivity capability is built into many automation components, typically via an Ethernet port, but can these connections be made secure?

Connections need to follow certain rules, and the IT department of a company should be involved in implementation. If connection to the IIoT is needed, the required infrastructure or service, such as a virtual private network (VPN) or a remotely hosted system, must be in place. A safe and secure connection costs more at the hardware level than a simple, less secure network, and although a great place to start, network security shouldn't stop there.

Security in Layers

Layered security, or defense in depth as the Department of Homeland Security refers to the concept, is a network defense strategy with multiple security layers to slowdown an attacker. There are many layers including physical access security, authorization and authentication, and data and communications encryption, with new technology layers being added.

Although not discussed here, don't forget about some of the other defense-in-depth strategies such as providing physical separation between corporate and control networks, employing DMZ zones, disabling unused ports and services, intrusion detection and antivirus software, and management of the upgrades and the remediation of security issues. All of these layers are typically planned out during the single-point failure analysis and the risk mitigation phase of the project, but let's get back to the physical access security layer.

Restricting physical access starts at the facility fence line, and continues all the way to the control cabinet or server room door. Placing the computers and controllers behind locked doors is a cost-effective way to reduce tampering and theft.

Authorization and Authentication

Another layer of security for IIoT connected networks is authorization and authentication. A variety of access controls are available to authenticate and restrict who has access to what data. Built-in tools are often available, in human machine interfaces (HMIs) and smartphones for example, enabling implementation of username and password policies. Role-based access controls can also be used, for example to keep operators out of engineering or technician functions and screens. Access lists provide security to a network by limiting access to only those users on a certain list. Access lists can also be used to decide which traffic is forwarded and which is blocked.

Authentication should be addressed at the HMI and hand held device level (smartphones and tablets), and with all plant floor PCs. Implementing a basic username and password requirement can restrict unauthorized users from accessing functionality and data in HMIs, databases and other plant floor applications.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Access control lists and application whitelisting is the next level, limiting access to sensitive data or network areas. Predefined accounts can be created to restrict each user's level of access to only what's necessary for him or her to perform their job. These accounts can be configured during the design stages of an HMI application, a task made easier when using modern HMIs with built in security and data encryption.

Network managers are typically responsible for implementing security features per industry standards and company policies. A good place to start is Homeland Security's National Cyber Security Division, Control Systems Security Program titled, Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies.

Learn More...Watch The Video!

C-more HMI Setup: Password Protect System Screen

http://n2adc.com/vdpq5

Remote Access Introduces Issues

Remote access presents security difficulties because information is now flowing into and out of the facility to users worldwide. While physical security and authentication are relatively easy ways to secure sensitive data, remotely accessing devices through a router and firewall is more complicated. This remote access is provided through another level of security which may include port forwarding, IP Security (IPSec) protocols and VPNs—all technologies that take some work to configure.

The port forwarding feature in routers and firewalls allows incoming messages from the IIoT to go to a specific device inside the facility. However, since port forwarding routs data between two devices, often over unknown networks, it opens a port to a hacker, so data could be intercepted and possibly changed.

The IPSec protocol closes up the port by providing authentication of the sender, and data encryption. It can be implemented at the end host or in routers. It's also useful for implementing VPNs for remote access to private networks.

A VPN is a reliable way to secure a remote access connection from a public network. It makes the remote device appear to be on the local network by giving the device an IP address on the network. This provides a secure remote access connection. However, these connections can be difficult to configure.

See "Using C-more Remote HMI App with secure VPN connection" section of Chapter 5 here.

Easy to Secure Layers

Security requirements don't change whether working locally or remotely. Fortunately, new technologies are making it easier to exchange data using the IIoT. There are several plug-and-play firewalls that are industrial protocol-aware, and many cloud providers provide excellent network security capability.

With the development of more powerful field devices, the weakest links are still authentication, authorization and secure communications. Some of the emerging technologies addressing these weaknesses include cloud access security broker (CASB) and software-defined networking (SDN)

CASB is middleware installed between a cloud service and a cloud user such as a smartphone, HMI or PC. To help secure internet and cloud-based applications, CASB manages and enforces enterprise security policies. Although CASB is not here yet, it's coming in the form of cybersecurity-as-a-service. This will make it easier to secure network communication with a diverse set of cloud providers connected to the local network.

Another new technology simplifying network security is SDN. SDN is similar to server, HMI and PC workstation virtualization. In SDN's case, it separates network configuration and security policy from the physical hardware and connections. The ability to create hardware instances virtually using software makes dynamic changes to network resources possible. Changes, expansion and reconfiguration become much quicker when network traffic and security are separated from the network hardware.

There are many more layers and new technologies to consider, as well as new security threats which will emerge. While one or two layers may eliminate most security threats, adding more layers will help eliminate them all.

System Integrators Talk HMI Projects

"Been there, done that," – besides just being a great t-shirt logo, there is a lot of truth in seeking wisdom from those who can claim they've been there and done that. In this "System Integrators Talk" segment, we queried our System Integrators Group on LinkedIn with a question on HMIs, and gathered their collective wisdom into one beneficial chapter.

What do you think are the biggest "gotcha" moments when designing an HMI project?

Here is a bullet list of some great takeaways from this conversation. Note: If you have the time, you should read the in-depth posts below for more details. These guys have a lot of experience and really gave some great responses.

- Before you get too fancy, first hook up your HMI to your PLC and make sure that you have your communications working. If it's not working, then the best HMI design will not matter.
- Talk to the actual operators. After all, if they are the ones using it, they are probably the ones who would have valuable input on how to optimize it.
- Watch the operators use your screens, and see if certain things are giving them trouble. Sometimes you will notice things that operators forget to tell you.
- Consider where the panel will be located. Observe its proximity to the user, as well as the angle of the screen.
- Remember that certain colors and symbols mean different things in different parts of the world. Red doesn't always mean STOP.
- Make sure that your HMI design gives the operator feedback as they use it. For example, if they push a button… make it change so they know that they pushed it.
- Make sure that messages that may be displayed on the HMI are correct and clear.
- Make sure that you have a well-defined scope that includes colors, alarms and screen examples before designing your panels. Get key people to sign off on your defined scope. This could save you a lot of rework and keep you from the dreaded scope creep.

For more details read the comments below…

Brian Gallogly the President of Quantum Automation…

First – get the communications taken care of – wire it up to the PLC, configure a pushbutton and make a Discrete Output point turn on, then put a potentiometer to an Analog Input point and have a bar go up and down on the screen. The worst gotcha is when you've done all the work but you can't get it to communicate.

Second – Talk to the person who will be operating the machine or process and get input from them as to how they would like the look and feel of the screens. Not talking to the ultimate operator can cause for multiple re-do's of the screens. Get the customer to sign off on the scope of work as to what the screens will look like, colors, alarms – who to email or text and when, reports – look and content, etc. before getting started. If you have example screens that have been printed out and put in a binder to show them, it cuts the time down tremendously.

Third – Having a spreadsheet of all the I/O with Engineering Units and Hi and Low Alarm limits on all the Analogs really saves time…. especially if you get your customer to sign off on the finalized spreadsheet prior to you configuring the screens.

Glenn Erickson the President at Expert Automation Design Inc…

Brian's input is very valid. If I was to add anything, watch the various operators use the touch screen and look for hesitation. If the operator is hesitating then they are not sure what to push or where to go. Ask them why they are hesitating and find a way to make the use of the screens more natural or intuitive.

Also, watch the operators eyes. If they are squinting, then the font or the text-size is inadequate or the contrast between the desired information and the background is too low.

I have found that a dark blue or light grey overall screen background are the most practical to use. Also, avoid the fancy pushbutton effects. They tend to obscure the text and function of the pushbutton.

Remember that 8% of all human males are color-blind in some combination of colors (Typically red and green.) Design your screens so that the change-of-state of a button or lamp is as dramatic as possible so that the user can be assured of the change in state, regardless of the colors they are actually seeing.

When displaying numerical or text data, I prefer a bright color like green in a black background surrounded by a frame that looks like a meter-face.

Be aware that different cultures often use different colors to indicate the state of a bit. Here in the U.S. we use green for ON and gray for OFF. In Europe, they use RED for ON and GREEN for OFF! So when developing screens for a customer in a different culture, ask for sample screens of another system they use that they like and imitate them.

Steve Myres, the Owner of Automation Control Solutions LLC…

I second what Brian says about prototyping various types of controls, especially when working with a new HMI or PLC or both. I've been called into customers who drew hundreds of controls on many screens, but didn't check to make sure they understood how to configure one before copying. It was easier to toss everything they had done and start over copying one working element than it would have been to make the corrections hundreds of times. That's what you avoid when you create and test one control before making all of them for the entire app.

Brian Woodley, a Control Strategy Specialist at Innotech Industrial Innovation Inc…

I think you've all covered the important points, well written. I always took the time to ensure my units and jargon were correct for the operators, the spatial layouts were accurate from the touch screen station viewpoint, and any cycle descriptions were accurate and truthful. Here's some minor additions to the list. Some are merely preferences, and doubtless most of you have your own ways of dealing with them.

I make sure to understand the exact location of the screen in relation to the view of the associated equipment and plan any animations or physical layouts accordingly. Pay attention to the scale and relative position of the equipment from that perspective, so when the operator looks up from the screen, it's a clean metaphor of what's in front of them. This forms an instant connection with the process and builds trust in the screen and what it is displaying. Graphical designs work great this way. I've seen so many text-based screens that operators learn to despise, as they are so disconnected from the process. Like looking at a spreadsheet instead of a dynamic operation.

I try not to put something on the screen or display a message unless it is true, with no qualifiers or conditions. I'd rather not display anything than provide misleading information. Since there's no scripting on the C-More screens, I sometimes I find it necessary to make dedicated internal coils for this purpose in the PLC that drive specific messages or combinations of situations. If the screen says it, the PLC thinks it, so when the display doesn't match reality troubleshooting can begin on the correct path. This means I don't implement things until they are verified, because if operators learn that some things on the screen are not always true, they can begin to doubt it all.

My primary HMI experience is with Wonderware, and you can have many contexts for the visibility of a display element or calculation of a value. I find that by processing things in the PLC and generating a special group of dedicated display bits you can have layered, sophisticated animations on the C-More that mimic Wonderware to some extent.

I also avoid using simple pushbuttons, always use indicator buttons, since most elements are copied and pasted for simplicity when building the pages – keeps the look and feel consistent. That way it's easier if the application requires a different status indication for certain buttons later on.

I like to show the operator they've hit a button even if the action they're requesting can't be taken for some interlocking reason. Poor feedback leads to frustration and can make a touch screen become what we call a "punch screen" as they press buttons harder and harder, as if that would cause something to happen.

This is easy to achieve with a small grey rectangle appearing around the button, visibility context of the button status being pressed. My indicator tag for a button is usually the action the button is trying to initiate, which also serves as a confirmation that the requested action is happening. If there's an interlock preventing the action, the operator will not see a change to the button when it is pressed. In that case the thin grey rectangle appears around the button as a clear indication they pressed it, and if the cycle or action does not occur most operators know and understand there's a reason for it. Then they can look into why the mode didn't activate instead of pounding on the screen.

One niggly little thing I often forget is to check the visibility context tab when copying and pasting. This is more a typo style of gotcha moment, one that is embarrassing if it makes it to the user and consumes of time re-working through the pages where it was missed, because it doesn't always show up during simulation.

That's all the thoughts for now. Good topic!

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Working with Objects

To create a functional HMI screen, simply place an object on a screen in the desired position, configure its appearance and needed actions, and link it to the appropriate tag(s). That's it in a nutshell. But what kind of objects are available and what actions they can take depends on your particular HMI. With C-more, there are over 50 different objects you can use to perform functions such as opening new screens, lighting indicators, metering, timing, bitmap animation, turning on and off bits, entering and displaying numeric values, and displaying alarms.

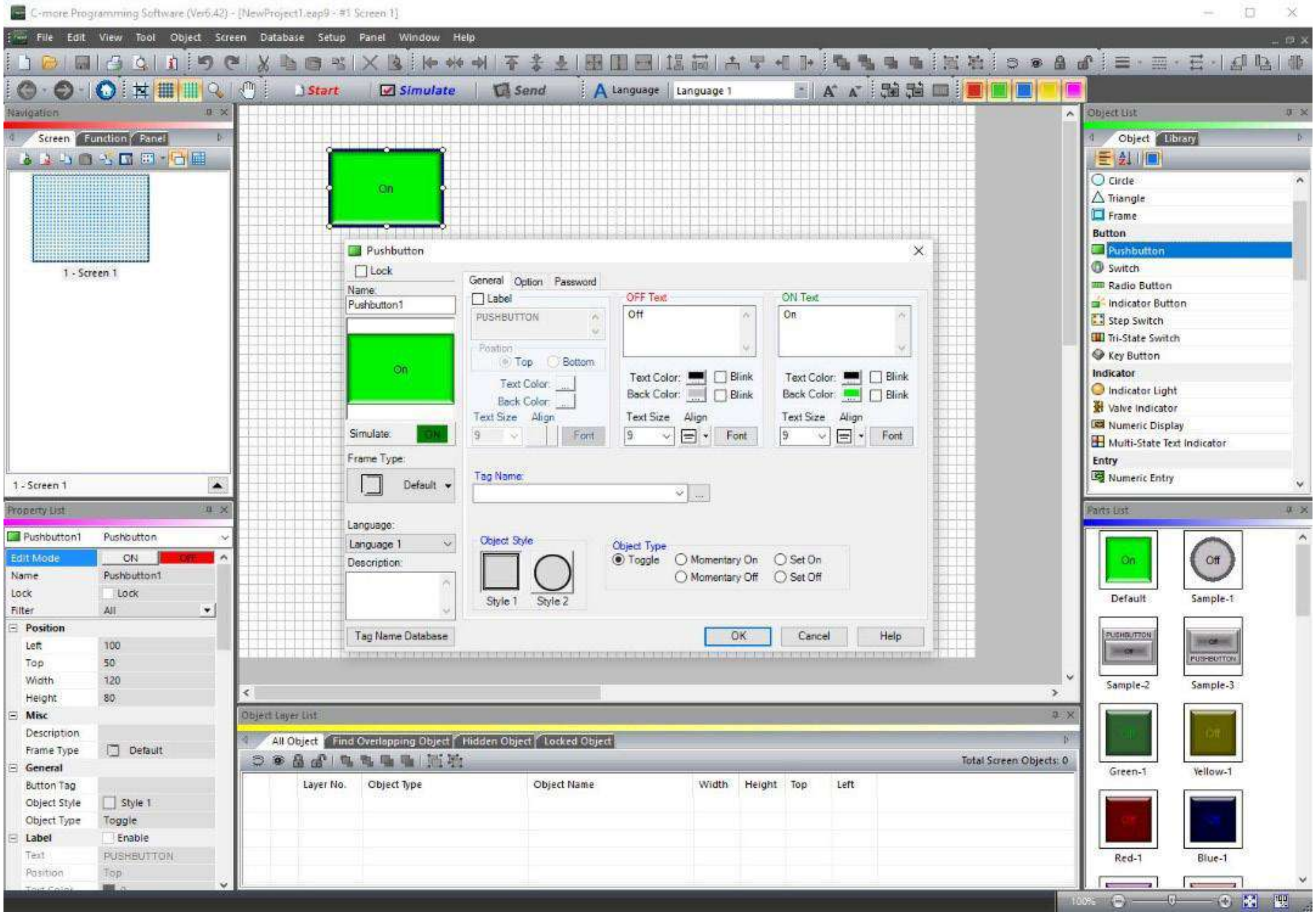
Objects available include:

- Shapes
- Buttons
- Indicators
- Entry Elements
  (numeric entry, increment/decrement, sliders, etc.)
- Meters/Graphs
- Bitmap Objects
- Text Elements
- Control/Navigation Objects
  (screen change pushbutton, screen selector, etc.)
- System Objects (volume control, set language, etc.)
- Recipes
- Alarm Lists
- Analog/Digital Clocks

Getting Started

Let's take a closer look at the options available when configuring an object. First, select an object and place it on a screen in the position you choose. This can be done with a simple drag-and-drop. You will then be able to set up many aspects of this object. As seen with the pushbutton selected below, you can assign a name, a tag, a language, a label, a description, choose the frame type, the object's style, text and background colors, set a sound, visibility, a press delay, make it blink, give it a password and more.

If you are not sure how a particular option will affect the appearance of the object, the "Simulate" selection will help. It's located below the object image and allows you to test the appearance of the object in different states. Depending on the object type, you can click on the ON or OFF box or enter a value to see the object in action.



Pushbutton

☐ Lock

Name: Pushbutton1

Simulate: ON

Frame Type: Default

Language: Language 1

Description:

Tag Name Database

General | Option | Password

☐ Label: PUSHBUTTON

Position: ● Top ○ Bottom

Text Color:

Back Color:

Text Size: 9 Align Font

OFF Text: Off

Text Color: ☐ Blink

Back Color: ☐ Blink

Text Size: 9 Align Font

ON Text: On

Text Color: ☐ Blink

Back Color: ☐ Blink

Text Size: 9 Align Font

Tag Name:

Object Style: Style 1 Style 2

Object Type: ● Toggle ○ Momentary On ○ Momentary Off ○ Set On ○ Set Off

OK Cancel Help

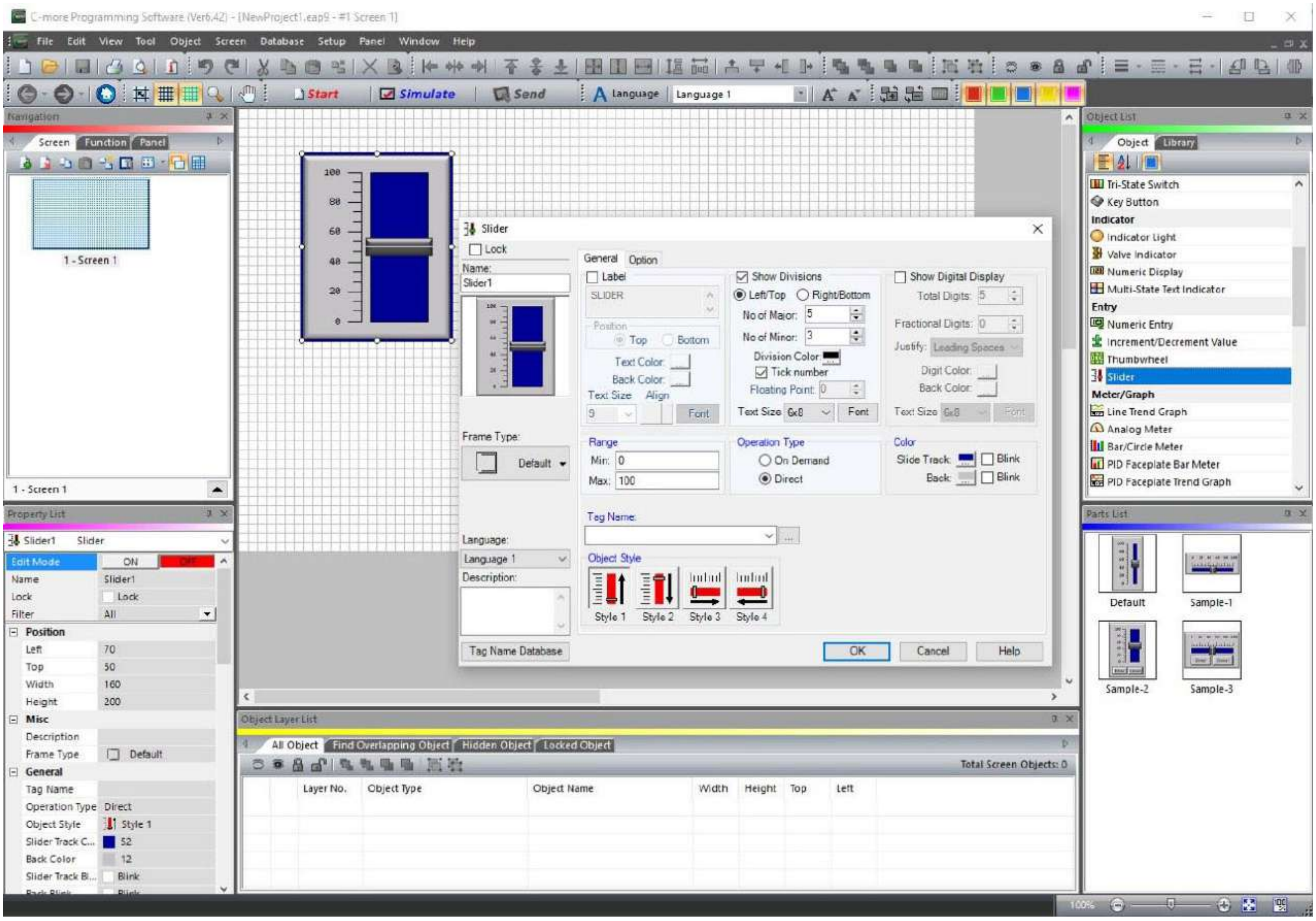
Property List

Pushbutton1 Pushbutton

Edit Mode	ON OFF
Name	Pushbutton1
Lock	☐ Lock
Filter	All
Position	
Left	100
Top	50
Width	120
Height	80
Misc	
Description	
Frame Type	Default
General	
Button Tag	
Object Style	Style 1
Object Type	Toggle
Label	☐ Enable
Text	PUSHBUTTON
Position	Top

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

If you were to choose a slider object, as seen below, the options would change to include the min/max range, division and tick mark setup, scaling options, etc.



Slider

☐ Lock

Name: Slider1

Frame Type: Default

Language: Language 1

Description:

General | Option

☐ Label
SLIDER
Position: ○ Top ○ Bottom
Text Color:
Back Color:
Text Size: 9 Align Font

☑ Show Divisions
● Left/Top ○ Right/Bottom
No of Major: 5
No of Minor: 3
Division Color:
☑ Tick number
Floating Point: 0
Text Size: 6x8 Font

☐ Show Digital Display
Total Digits: 5
Fractional Digits: 0
Justify: Leading Spaces
Digit Color:
Back Color:
Text Size: 6x8 Font

Range
Min: 0
Max: 100

Operation Type
○ On Demand
● Direct

Color
Slide Track: ☐ Blink
Back: ☐ Blink

Tag Name:

Object Style: Style 1, Style 2, Style 3, Style 4

Tag Name Database | OK | Cancel | Help

Property List: Slider1 Slider

Edit Mode	ON OFF
Name	Slider1
Lock	☐ Lock
Filter	All
Position	
Left	70
Top	50
Width	160
Height	200
Misc	
Description	
Frame Type	Default
General	
Tag Name	
Operation Type	Direct
Object Style	Style 1
Slider Track C...	52
Back Color	12
Slider Track Bl...	Blink

Object Library

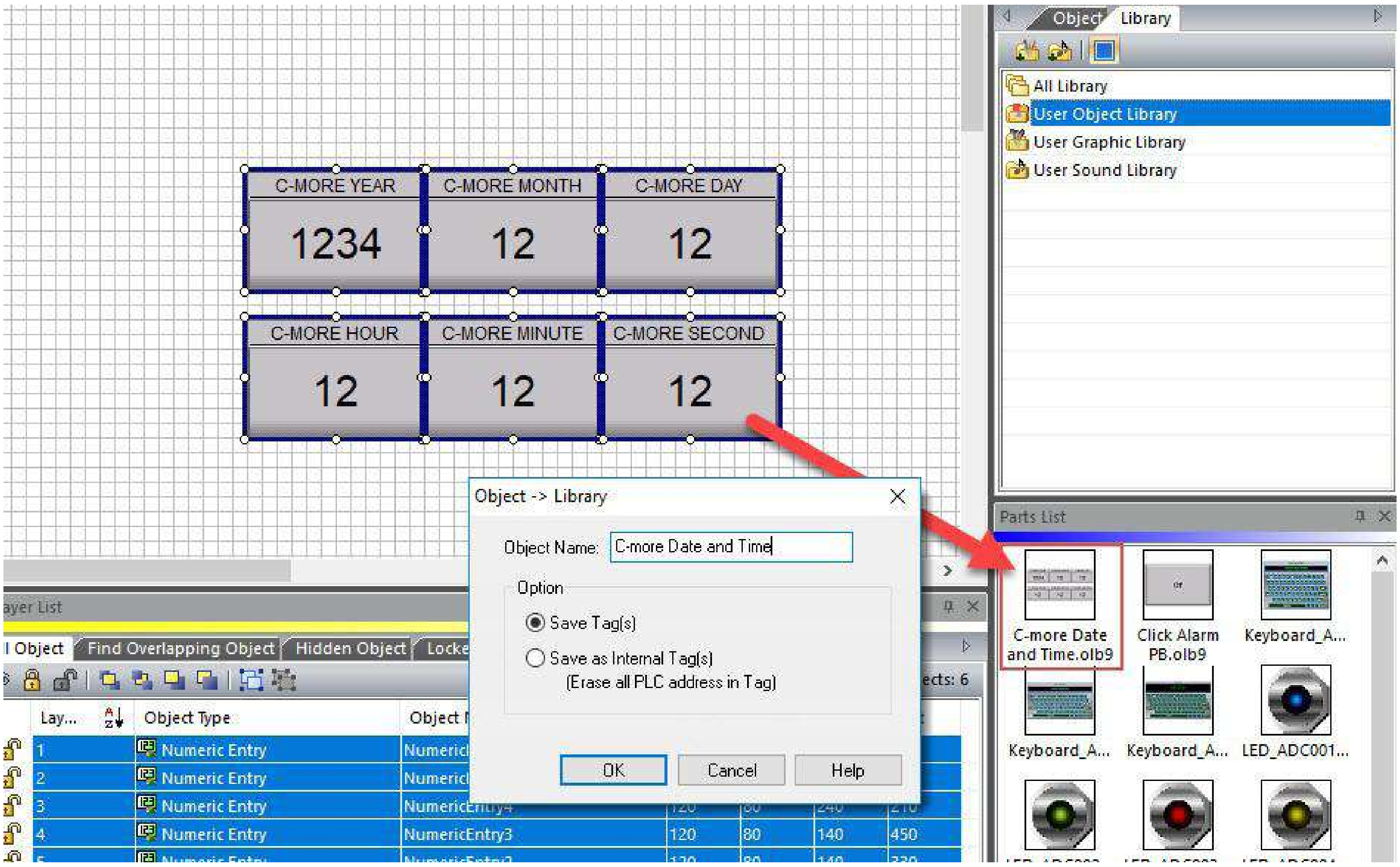
Great, you have created an object you need for this project. But what if that object could also be used in numerous upcoming projects? Do you really want to take the time to create it over and over again? Of course not, and that's where the Object Library comes in very handy.

The Object Library allows you to save common objects so that you don't have to create them repeatedly for each subsequent project. Once you've created the object you wish to save, choose the "Object->Library…" option from the right-click menu. You will then be given the option to name it and save the tags associated with it. Choosing "Save Tags" will save the PLC Address(es) to the Tag Name(s) in the object. When the object is added to another project later, the tags will be added automatically into the Tag Name Database. If there is no device that supports the PLC Addresses assigned to the tags, a device will be added automatically to the Panel Manager.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

To make all tags associated with the object internal, click the Save as Internal Tag(s) radio button. Any PLC Address(es) will be erased. When the object is added to another project later, the tags will be added automatically into the Tag Name Database. The tag will then need to be associated with a Device and a PLC Address assigned to it.

After the object is saved it will appear in the "Parts List" allowing for an easy drag-and-drop whenever it is needed.



Object -> Library

Object Name: C-more Date and Time

Option

● Save Tag(s)

○ Save as Internal Tag(s)
(Erase all PLC address in Tag)

OK Cancel Help

C-more also has libraries for saving screens, sounds, and graphic items for future use and these libraries can be exported and imported to other devices if needed.

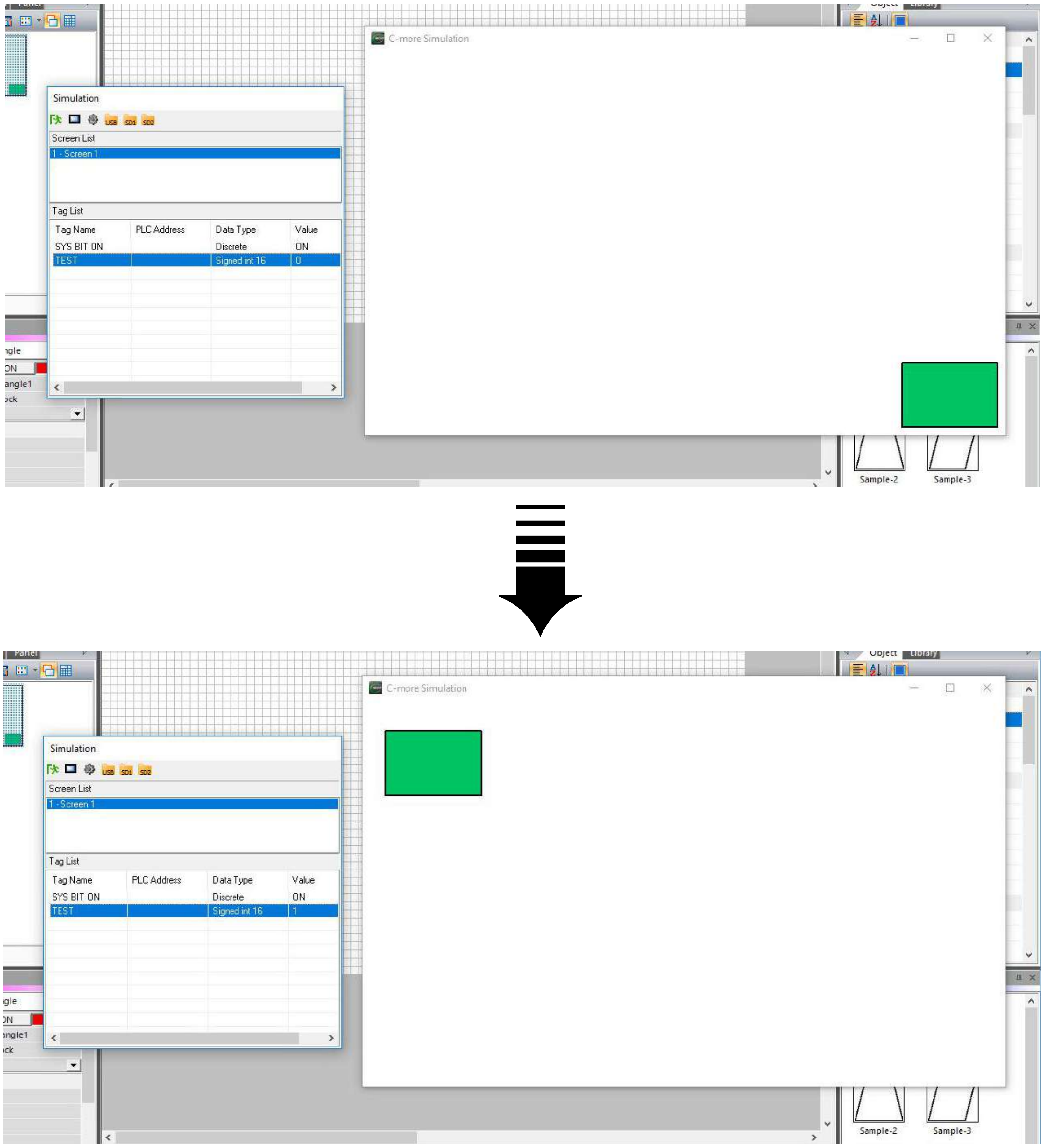
Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Object Animation

Most objects allow for color and visibility changes, but there are some objects that can also be animated with position moves, sizing and rotation. The C-more Programming Software allows object move, size and rotation animation with line, rectangle, circle, triangle and animation bitmap objects.

Types of animations:

- **Move** – object will move along the X/Y axis (axis animation) or to multiple points designated by the user (point animation - shown below)



Simulation

Screen List

1 - Screen 1

Tag List

Tag Name	PLC Address	Data Type	Value
SYS BIT ON		Discrete	ON
TEST		Signed int 16	0

Simulation

Screen List

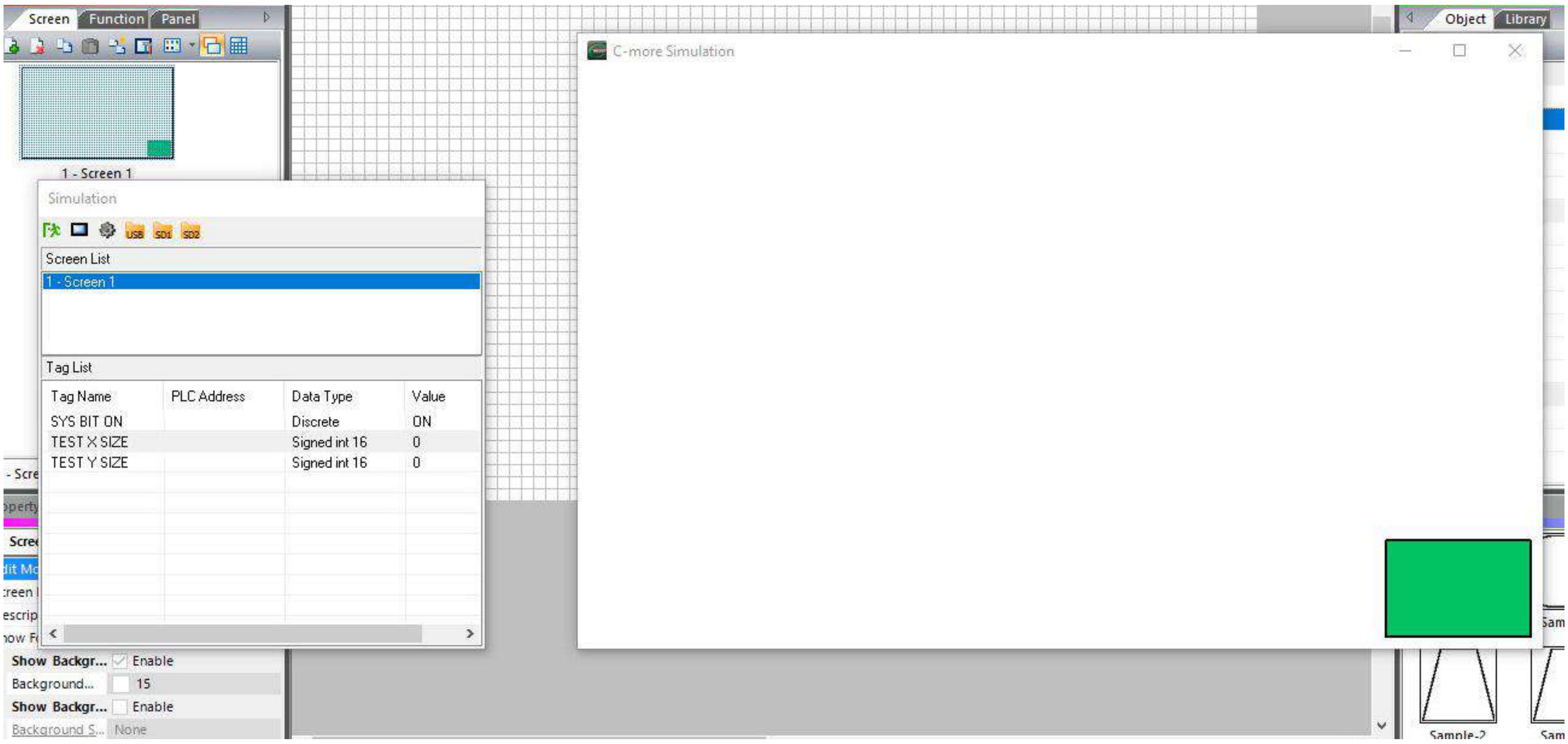
1 - Screen 1

Tag List

Tag Name	PLC Address	Data Type	Value
SYS BIT ON		Discrete	ON
TEST		Signed int 16	1

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

- **Size** - these settings will cause the object to expand or contract vertically and/or horizontally

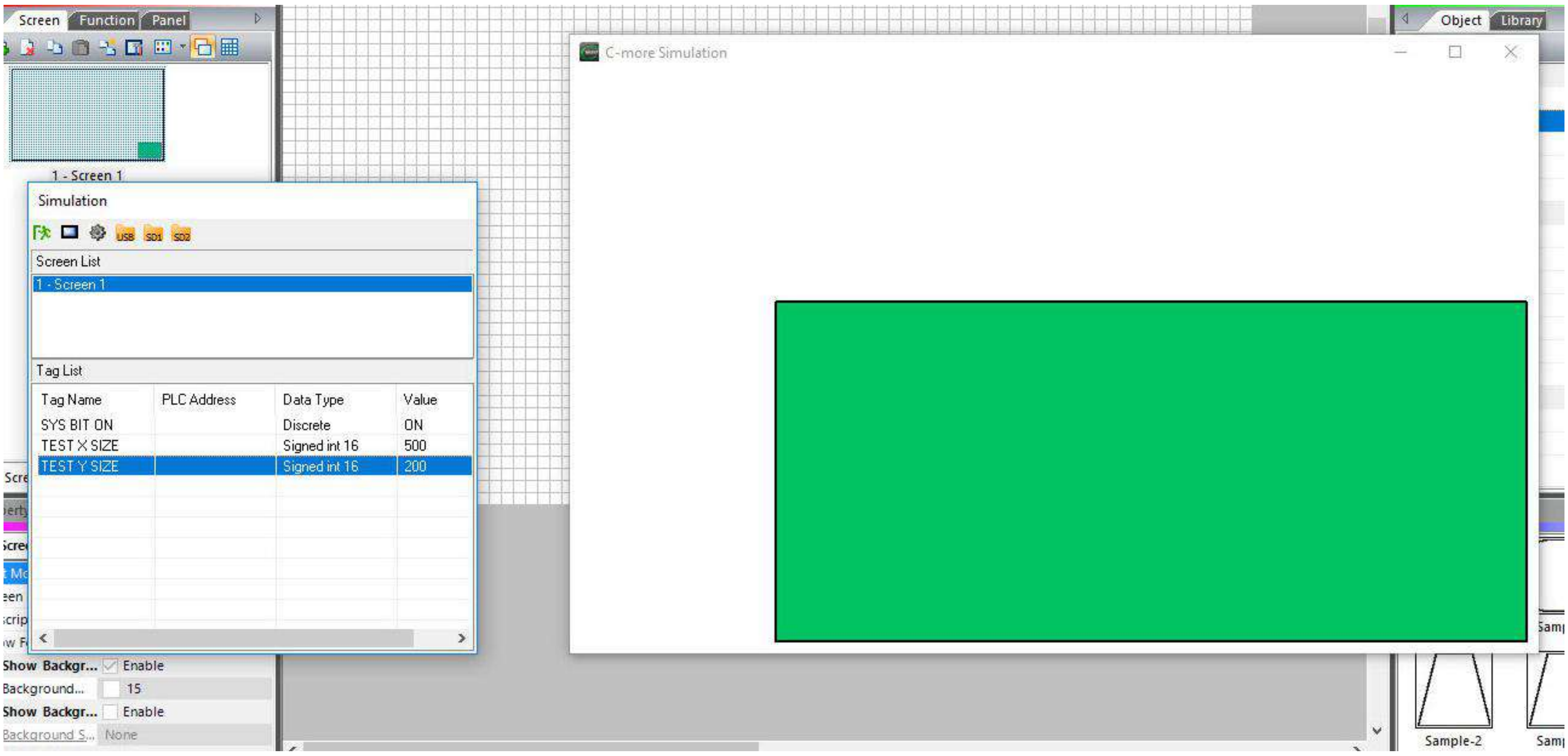


Screen List

1 - Screen 1

Tag List

Tag Name	PLC Address	Data Type	Value
SYS BIT ON		Discrete	ON
TEST X SIZE		Signed int 16	0
TEST Y SIZE		Signed int 16	0



Screen List

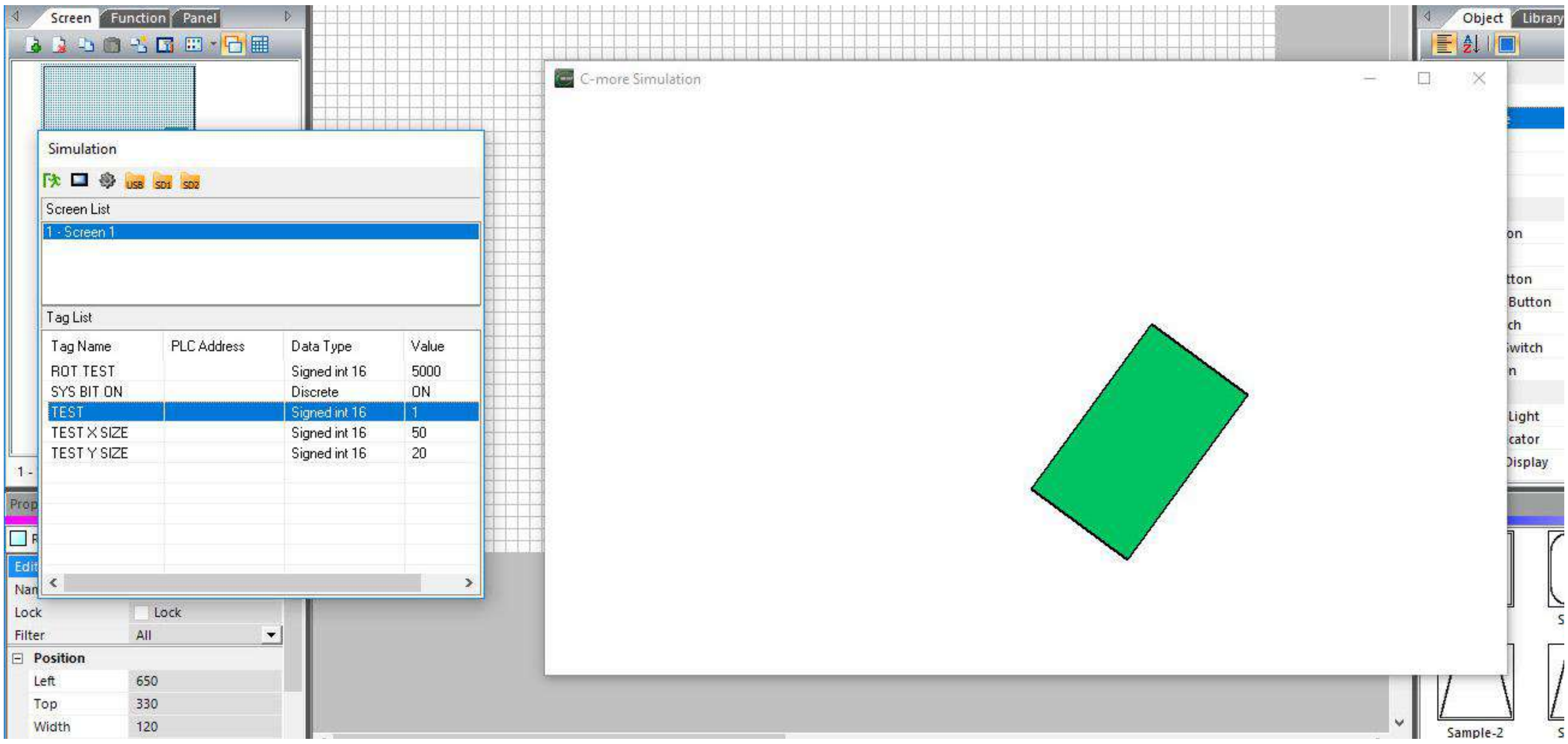
1 - Screen 1

Tag List

Tag Name	PLC Address	Data Type	Value
SYS BIT ON		Discrete	ON
TEST X SIZE		Signed int 16	500
TEST Y SIZE		Signed int 16	200

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

- **Combinations** – with C-more you can combine move, size and rotation animations into a more complex object



C-more Simulation

Simulation

Screen List

1 - Screen 1

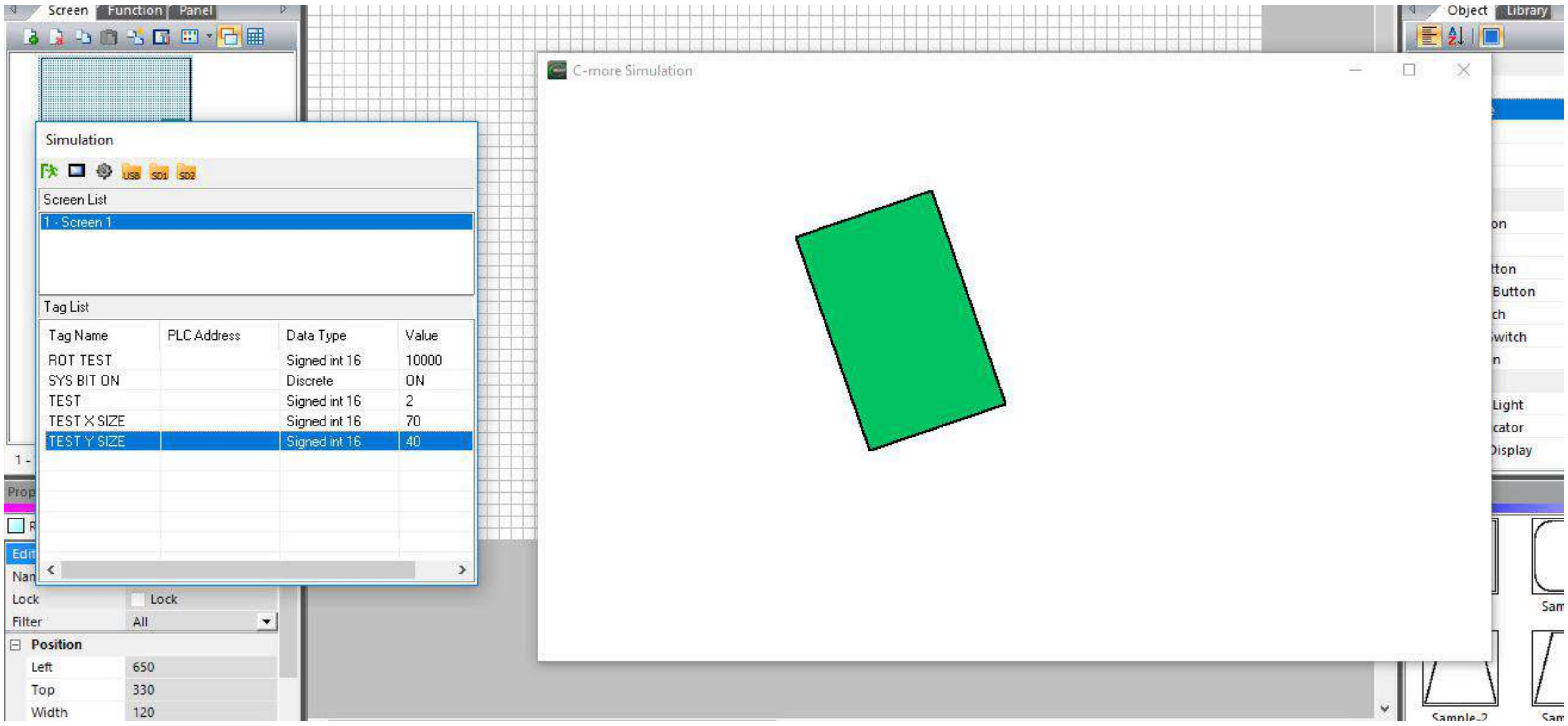
Tag List

Tag Name	PLC Address	Data Type	Value
ROT TEST		Signed int 16	5000
SYS BIT ON		Discrete	ON
TEST		Signed int 16	1
TEST X SIZE		Signed int 16	50
TEST Y SIZE		Signed int 16	20

Lock □ Lock
Filter All

Position

Left	650
Top	330
Width	120



C-more Simulation

Simulation

Screen List

1 - Screen 1

Tag List

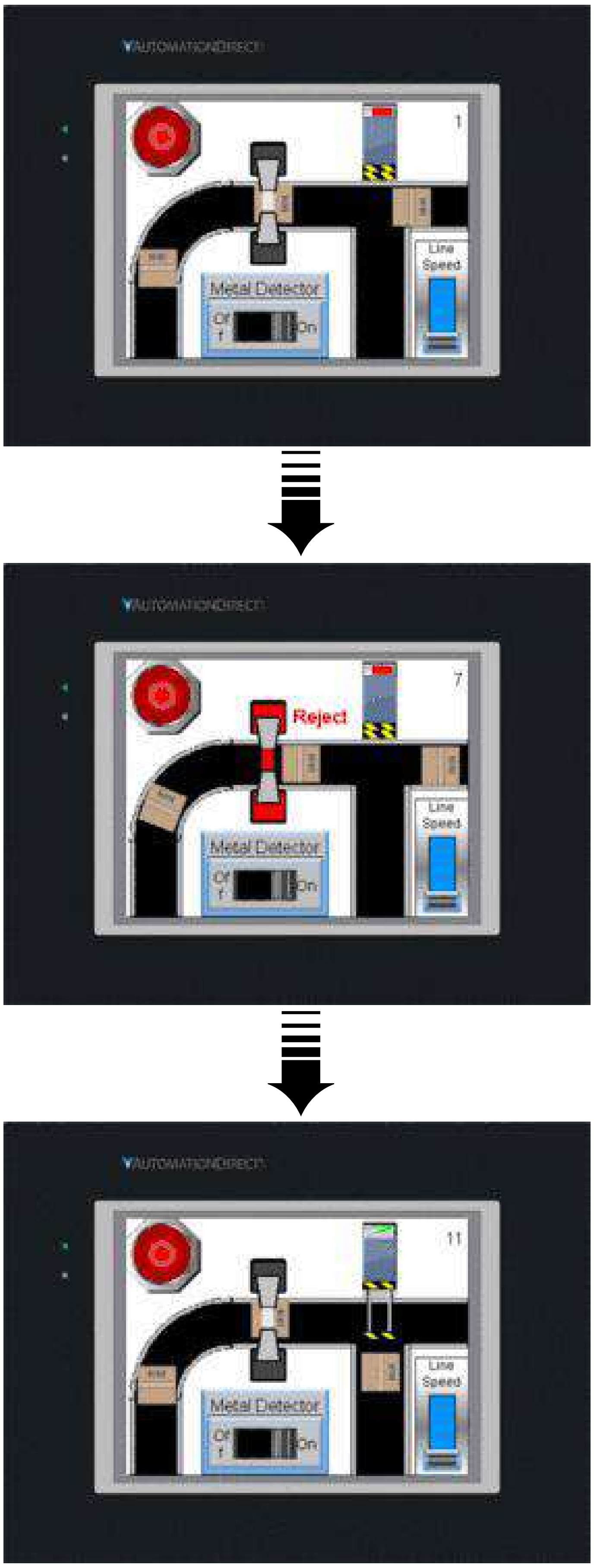
Tag Name	PLC Address	Data Type	Value
ROT TEST		Signed int 16	10000
SYS BIT ON		Discrete	ON
TEST		Signed int 16	2
TEST X SIZE		Signed int 16	70
TEST Y SIZE		Signed int 16	40

Lock □ Lock
Filter All

Position

Left	650
Top	330
Width	120

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs



Objects are the building blocks for HMI screen design. Understanding the objects and the options available with them in your HMI package is the key to creating an effective operator interface.

To learn more about C-more HMIs and what they offer, head on over to our store at www.automationdirect.com/c-more. The C-more Programming Software Help File has a wealth of information on configuring objects and animations. You can view that help file by downloading the latest version at https://support.automationdirect.com/products/cmore.html

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Events and Event Actions

The Basics

Just like any programmable device, an HMI needs to be told when to react and how to react. Objects on a screen can change when pressed or when their assigned tag values change. But what about the times when you would prefer something to happen based on a different occurrence, like time of day? Well, the C-more programming software does provide a way for the HMI itself to perform tasks based on conditions like these. In C-more these are called Events and Event Actions.

Events

There are three types of events in the C-more software:

1. Tag Event
2. Day/Time Event
3. Screen Change Event

The Tag Event is used to cause actions based on the condition of a tag or combination of tags. Tags used to trigger an event can be discrete, numeric or a combination of both. Discrete type tags will trigger an action based on the tag's transition from one state to another. Up to 4 discrete tags are allowed to control an action using different combinations of AND and OR logic. A numeric tag can trigger an action based upon limits reached by the value of the tag.

The Day/Time Event type is used to schedule an exact time that you want an action to occur. The scheduled event can be set up to occur every day or specific days of the week and at specific times.

The Screen Change Event is used to cause an action(s) to occur when a particular screen is changed, either opened or closed.

Event Actions

So now that we understand what events are possible with C-more, let's look at the actions that go along with them. C-more HMI's provide 11 different event actions:

1. Alarm
2. Tag
3. Tag Copy
4. Sound
5. Screen Change
6. Screen Capture
7. Mail
8. FTP
9. Backlight
10. Message Box
11. Math

The Alarm action only works with the Tag Event and will display the configured Alarm Text in the Alarm Banner at the bottom of the C-more panel screen when the Event is triggered.

The Tag action will set the value of a tag to a preconfigured value or a high/low state.

The Tag Copy action copies the tag value from the Source Tag to the Destination Tag. This action can be used effectively to copy data from one controller to another, allowing the C-More to act as a gateway between 2 devices on dissimilar networks.

The Sound action allows you to specify the internal beeper to "beep" or a sound file to be played through the audio line output port based on the occurrence of an event.

The Screen Change action switches the currently displayed screen to the specified screen when the event is triggered.

The Screen Capture action gives you the ability to capture an image of any of the touch panel's screens, either the screen that is currently displayed at the instant the event takes place, or any of the other screen.

The Mail action allows you to send specified Mail (email) messages with attachments to addresses through a specified Mail server.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

The FTP action allows you to send alarm logs, message database, trend data logs and screen capture files from the panel to a specific FTP server.

The Backlight action is used to control the touch panel's backlight based on an event. The panel's backlight can be either turned OFF or turned ON.

The Message Box action will display the specified pop-up message in the language you choose and an icon to indicate the type of message when the event is triggered.

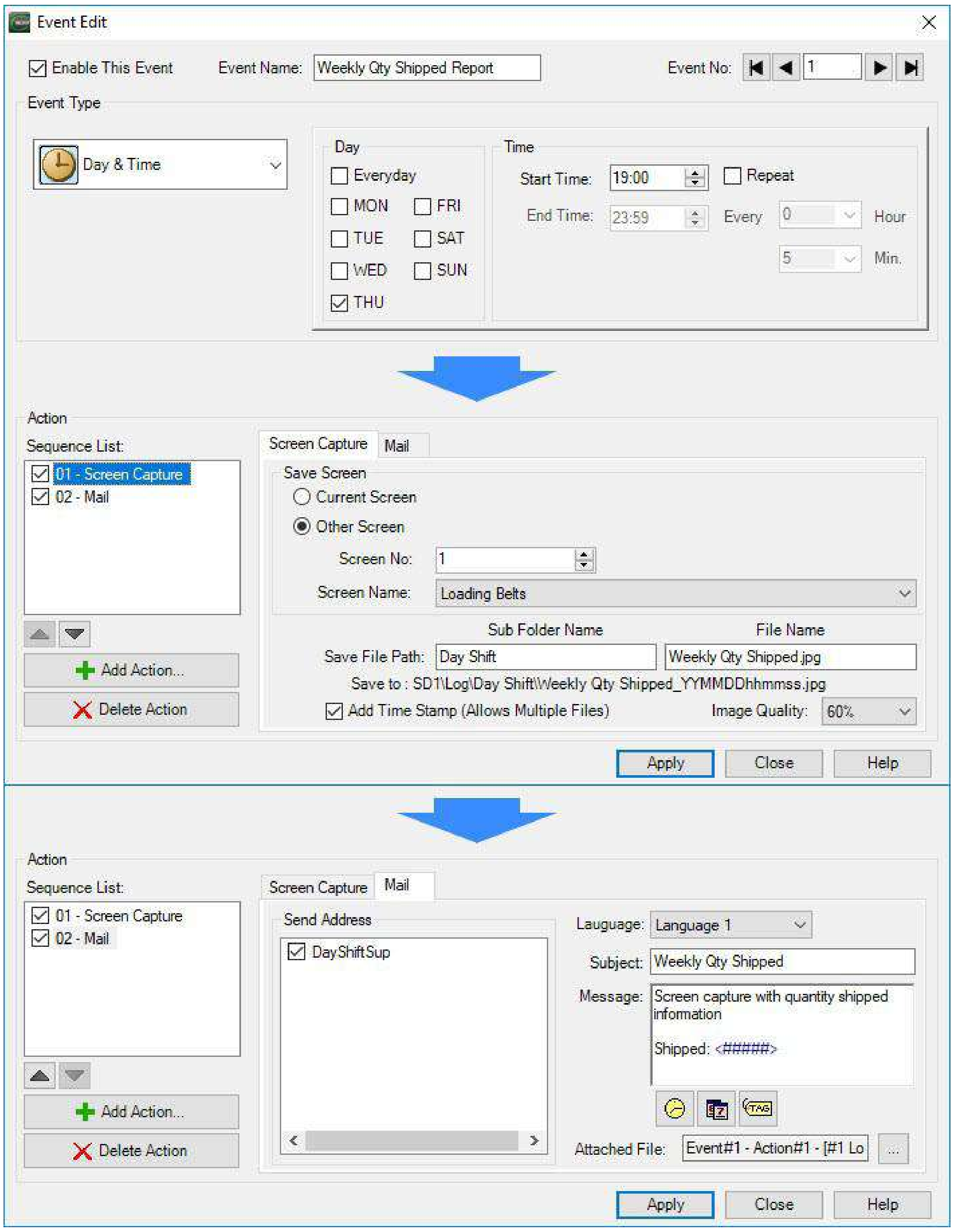
The Math action solves a user-defined formula when the condition defined in the event transitions to TRUE.

Event in Action

Now let's use some of these events and actions in real world examples.

Example 1

Your product shipments go out every Thursday at 6pm. The Loading Belt PLC is tracking the amount of product being loaded on the trucks and that information is displayed on a C-more HMI. Example 1 below shows the C-more HMI's Event Manager configured to send the quantity shipped information via email (with screen capture and embedded data) every week to the appropriate personnel. Notice the Day & Time Event is the trigger for the Screen Capture and Email Actions that are completed in succession.



Event Edit

☑ Enable This Event　Event Name: Weekly Qty Shipped Report　Event No: 1

Event Type: Day & Time

Day: □ Everyday　□ MON　□ FRI　□ TUE　□ SAT　□ WED　□ SUN　☑ THU

Time: Start Time: 19:00　□ Repeat　End Time: 23:59　Every 0 Hour　5 Min.

Action

Sequence List: ☑ 01 - Screen Capture　☑ 02 - Mail

Screen Capture

Save Screen: ○ Current Screen　● Other Screen

Screen No: 1

Screen Name: Loading Belts

Save File Path: Sub Folder Name: Day Shift　File Name: Weekly Qty Shipped.jpg

Save to : SD1\Log\Day Shift\Weekly Qty Shipped_YYMMDDhhmmss.jpg

☑ Add Time Stamp (Allows Multiple Files)　Image Quality: 60%

Mail

Send Address: ☑ DayShiftSup

Lauguage: Language 1

Subject: Weekly Qty Shipped

Message: Screen capture with quantity shipped information

Shipped: <#####>

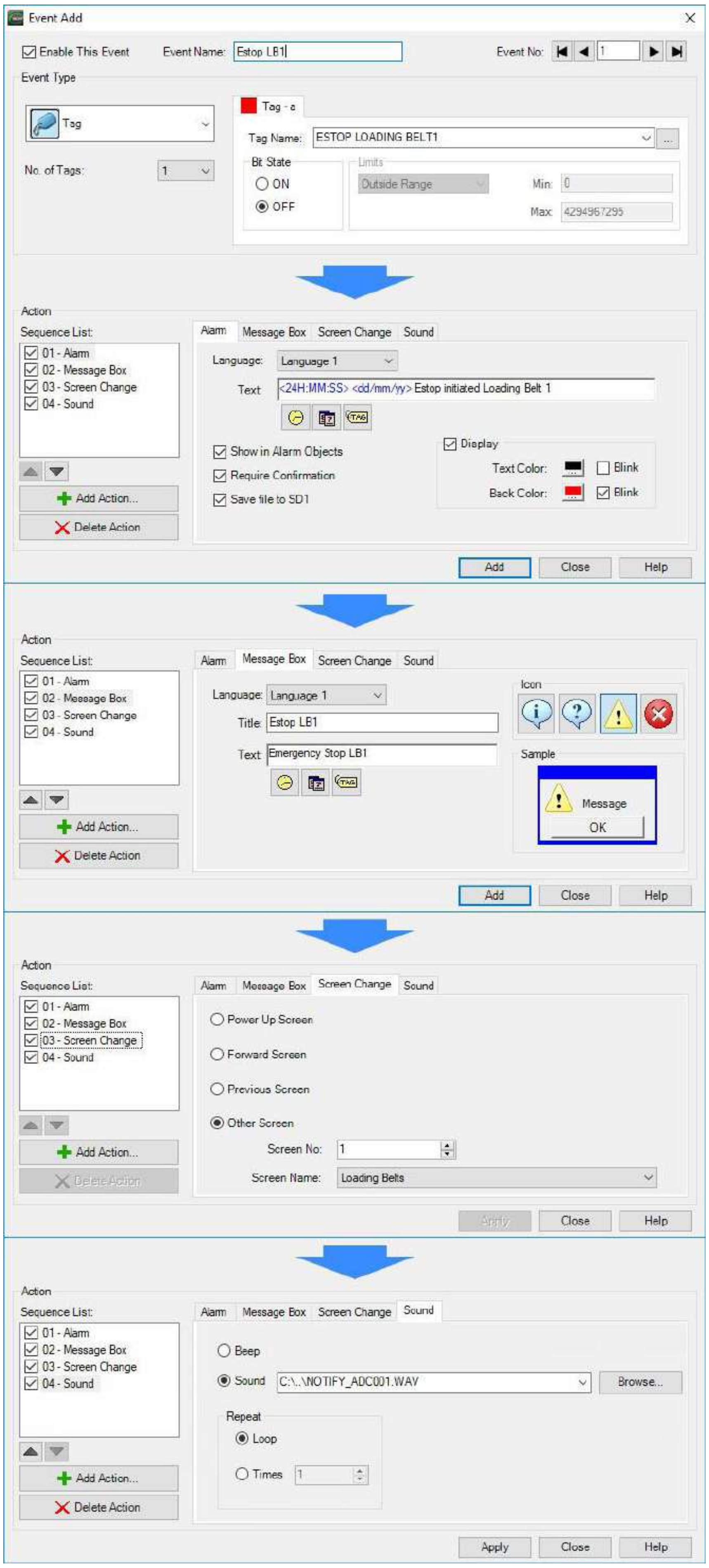
Attached File: Event#1 - Action#1 - [#1 Lo

Example 1

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Example 2

Same production facility but this time an Estop is activated on the loading belt. Example 2 demonstrates how the Event Manager can be set up to react to this situation. With this setup, the HMI will display a blinking alarm message in the alarm banner along with a message box notifying the operator of the condition, it will automatically change screens to the screen showing where the Estop is located, and the HMI will play a prerecorded audible alert message.



Event Add

☑ Enable This Event
Event Name: Estop LB1
Event No: 1

Event Type: Tag
No. of Tags: 1
Tag - a
Tag Name: ESTOP LOADING BELT1
Bit State: ○ ON ● OFF
Limits: Outside Range Min: 0 Max: 4294967295

Action
Sequence List:
☑ 01 - Alarm
☑ 02 - Message Box
☑ 03 - Screen Change
☑ 04 - Sound

Alarm
Language: Language 1
Text: <24H:MM:SS> <dd/mm/yy> Estop initiated Loading Belt 1
☑ Show in Alarm Objects
☑ Require Confirmation
☑ Save file to SD1
☑ Display
Text Color: □ Blink
Back Color: ☑ Blink

Message Box
Language: Language 1
Title: Estop LB1
Text: Emergency Stop LB1

Screen Change
○ Power Up Screen
○ Forward Screen
○ Previous Screen
● Other Screen
Screen No: 1
Screen Name: Loading Belts

Sound
○ Beep
● Sound C:\..\NOTIFY_ADC001.WAV
Repeat
● Loop
○ Times 1

Example 2

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Understanding Recipes

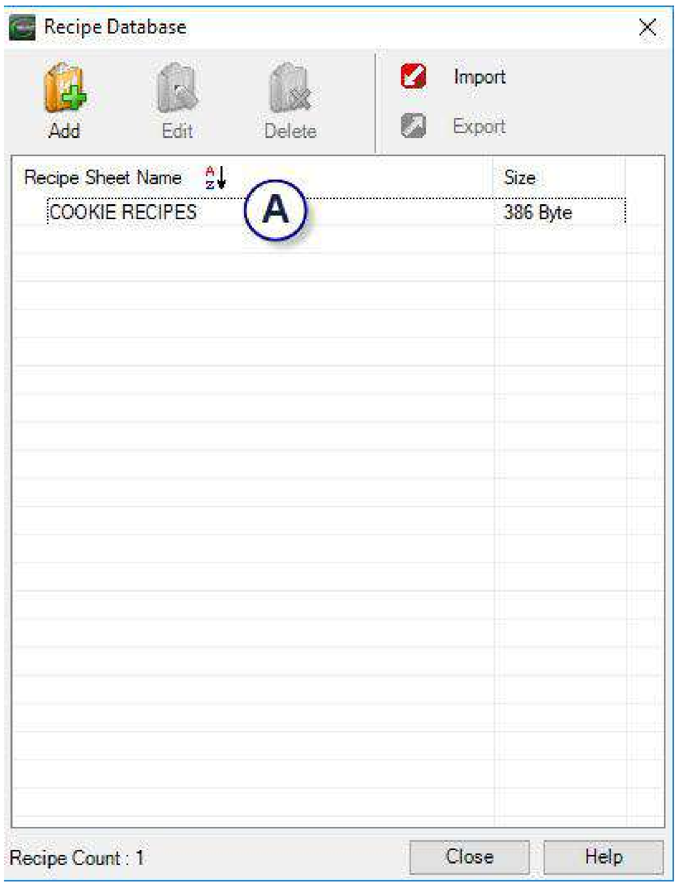
Recipes are used in a multitude of control applications including batch processing for the food, beverage, chemical and pharmaceutical industries, as well as complex applications such as robotics. Recipes, when used with an HMI, provide a simple and effective way to change the data in a large number of tags with the push of a single button.

The C-more touch panel has an advanced recipe function that allows operators to easily view and select from a list of multiple recipes. The operator simply touches a Call Recipe button on the screen, a Recipe Sheet pops up with a list of recipes, and the operator chooses which recipe to run. If configured so, the operator can even change values in a recipe from the HMI screen.

Recipe Database

The Recipe Database, found under the Database drop-down menu or Function tab, can store up to 99 user defined recipe sheets, with each recipe sheet configurable with up to 1,000 recipes. A single recipe can have up to 256 tags or values.

Creating a Recipe Sheet



Recipe Sheet Name	Size
COOKIE RECIPES	386 Byte

Figure 1

We will create a recipe sheet called "Cookie Recipes" (Figure 1, item A). When creating a recipe sheet, a spreadsheet style window is used (Figure 2). The first column in the sheet contains the recipe name destination tag (Figure 2, item A) and the names of the recipes.

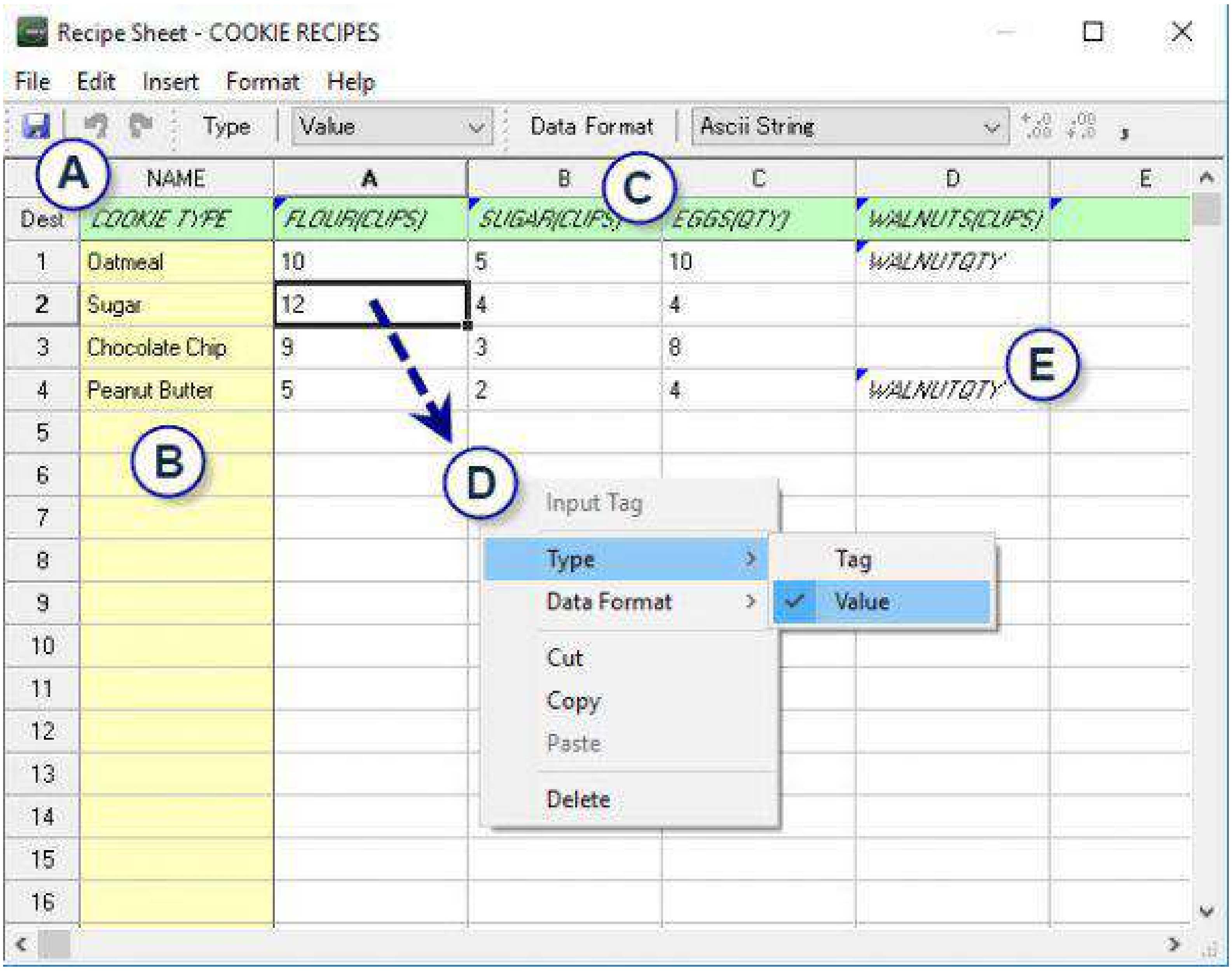
In this example we are setting up cookie recipes, so we created a tag called COOKIE TYPE for our recipe tag name destination. The destination can be an internal C-more tag or a PLC tag using an ASCII String data type. We used a PLC tag so the PLC can track which cookie recipe is being used.

Below the COOKIE TYPE tag we listed our various cookies: Oatmeal, Sugar, Chocolate Chip and Peanut Butter (Figure 2, item B). These names are static text and are not associated with any tags. When an operator selects and loads one of the cookie recipes, the cookie name is loaded into the COOKIE TYPE tag (Figure 2, item A), which can be displayed on the C-more panel, or used for control or logging data in the PLC or panel.

Next we added the ingredient tags: Flour(Cups), Sugar(Cups), Eggs(Qty), and Walnuts(Cups) (Figure 2, item C). The ingredients are the destination tags where the values from the recipe are loaded. These are entered in the green cells at the top of each column. They can be Tag Names that were previously created in the Tag name Database, or a new Tag Name can be created by simply entering the desired Tag Name in a cell and a dialog box will open for configuration.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Since different cookie recipes typically call for different quantities of ingredients, we needed to list how many units of each ingredient are used for each cookie type. By right-clicking on a cell in one of the ingredient columns we can select the cell type as a value or a tag (Figure 2, item D). If selecting a value type cell, we enter a number that is within the range of the data format, such as BCD, decimal, etc. If a sugar cookie uses 12 cups of flour, we enter 12 in the flour cell for the sugar cookie recipe. The values will be written to the ingredient destination tags when an operator selects a cookie type and presses the LOAD button.



Recipe Sheet - COOKIE RECIPES

Type: Value   Data Format: Ascii String

	NAME	A	B	C	D	E
Dest	COOKIE TYPE	FLOUR(CUPS)	SUGAR(CUPS)	EGGS(QTY)	WALNUTS(CUPS)	
1	Oatmeal	10	5	10	WALNUTQTY	
2	Sugar	12	4	4		
3	Chocolate Chip	9	3	8		
4	Peanut Butter	5	2	4	WALNUTQTY	
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Figure 2

If using a tag type cell, we assign a tag name, such as WALNUT QTY (Figure 2, item E), and assign a PLC address. This way the PLC knows the amount of walnuts used for a specific cookie, such as Chocolate Chip cookies.

Some other facts to know about the recipe sheet are:

- A blue triangle in the upper left corner of a cell denotes the cell is a tag type, no triangle is a value type
- If incorrect data is entered in a cell it will be displayed in red
- The recipe sheet can be edited using drag and copy/autofill features similar to an Excel™ spreadsheet
- The recipe sheet can be exported to and imported from Excel as an XLS or CSV file for easy editing, using care to maintain the syntax and format, or a CSV file may be accessed through the C-more panel FTP server feature

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Creating the Call Recipe Button Object

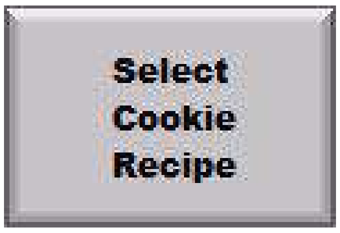
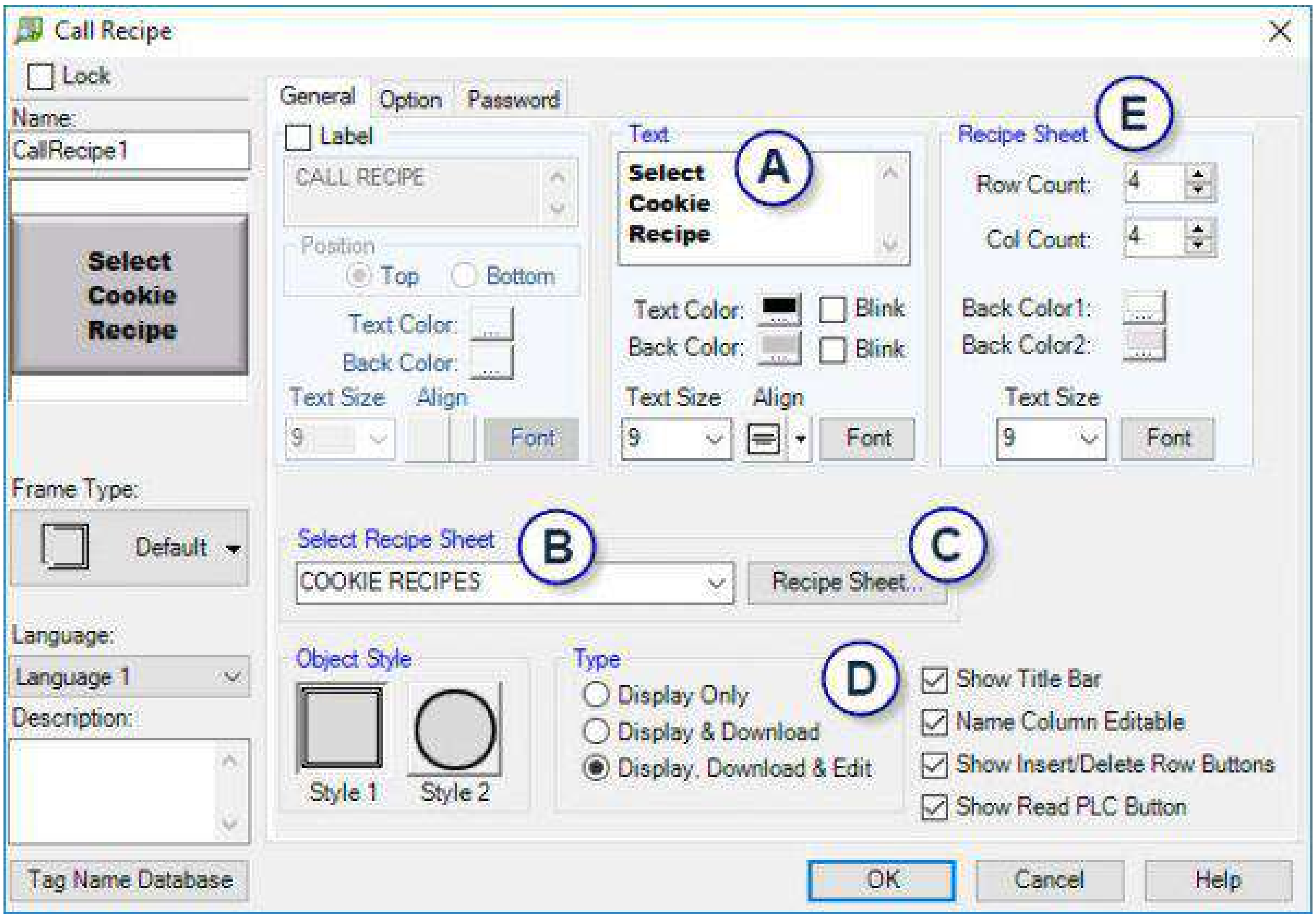


Figure 3

The Call Recipe button is used to display a specific recipe sheet on the panel (Figure 3). The button can be placed on one or more screens of a C-more panel. When an operator selects the button, the recipe sheet is displayed on the panel screen.

To configure the Call Recipe button, select it from the objects list. Enter the text to be displayed on the button (Figure 4, item A). We called ours “Select Cookie Recipe”. Next, select “Cookie Recipes” from the drop down list of Recipe Sheets (Figure 4, item B). To view and edit a recipe sheet, select the “Recipe Sheet” button (Figure 4, item C).

When configuring the Call Recipe button object, there are three types of sheets to choose from: Display Only, Display and Download, and Display, Download and Edit (Figure 4, item D). 1) Display Only will not allow any input from the operator. 2) Display and Download will allow the operator to select a recipe and load it into the destination tags. 3) The third type will allow the operator to edit the value in a cell that is to be written to a destination tag. Since we want to allow the operator to change the amount of ingredients in the recipes, we selected the third type.


Call Recipe

□ Lock

Name: CallRecipe1

Frame Type: Default

Language: Language 1

Description:

General | Option | Password

□ Label: CALL RECIPE

Position: ● Top ○ Bottom

Text: Select Cookie Recipe

Text Color: □ Blink

Back Color: □ Blink

Text Size: 9

Recipe Sheet

Row Count: 4

Col Count: 4

Back Color1:

Back Color2:

Text Size: 9

Select Recipe Sheet: COOKIE RECIPES

Object Style: Style 1 | Style 2

Type:
○ Display Only
○ Display & Download
● Display, Download & Edit

☑ Show Title Bar
☑ Name Column Editable
☑ Show Insert/Delete Row Buttons
☑ Show Read PLC Button

Figure 4

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

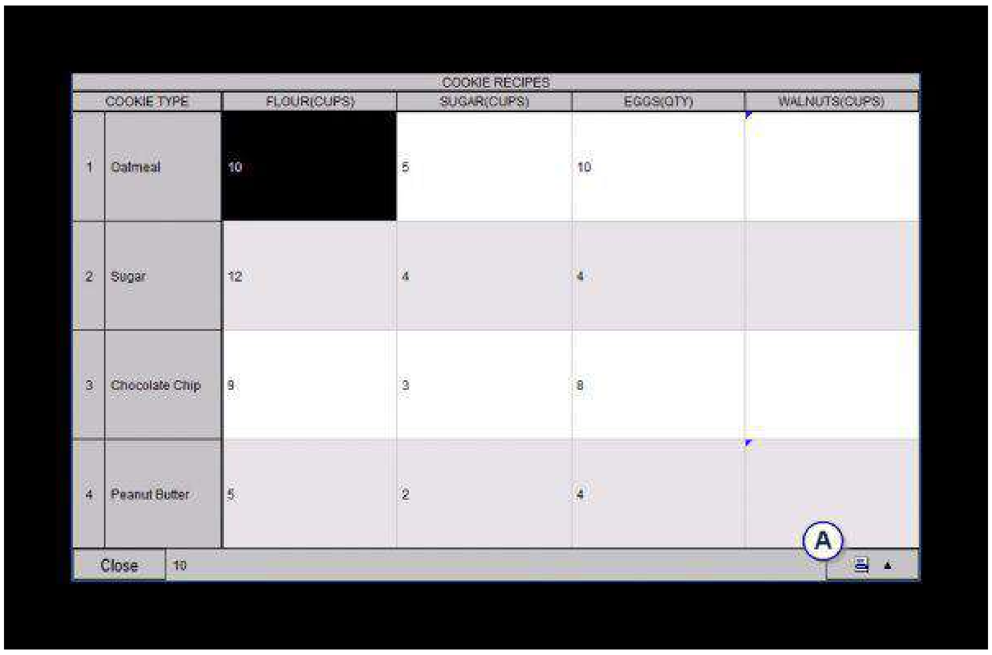
The Call Recipe button object also allows predetermination of how many recipe sheet columns and rows will be displayed on the panel screen (Figure 4, item E). Use the Simulate feature to test the project for the optimum display. In this case some of the ingredient names are rather long, therefore we will set the number of columns to four in order to view the entire name. We have four types of cookies, so we will select to display four rows.

Other features of the Call Recipe button include adjusting text size and changing fonts for the button text and for the recipe sheet text.

If preferred, a "Recipe Button" object can be used to load a single recipe from any recipe sheet.

Using the Recipe Function on the C-more Panel

Once the project is loaded into the panel, simply touch the "Select Cookie Recipe" button to display the recipe screen.



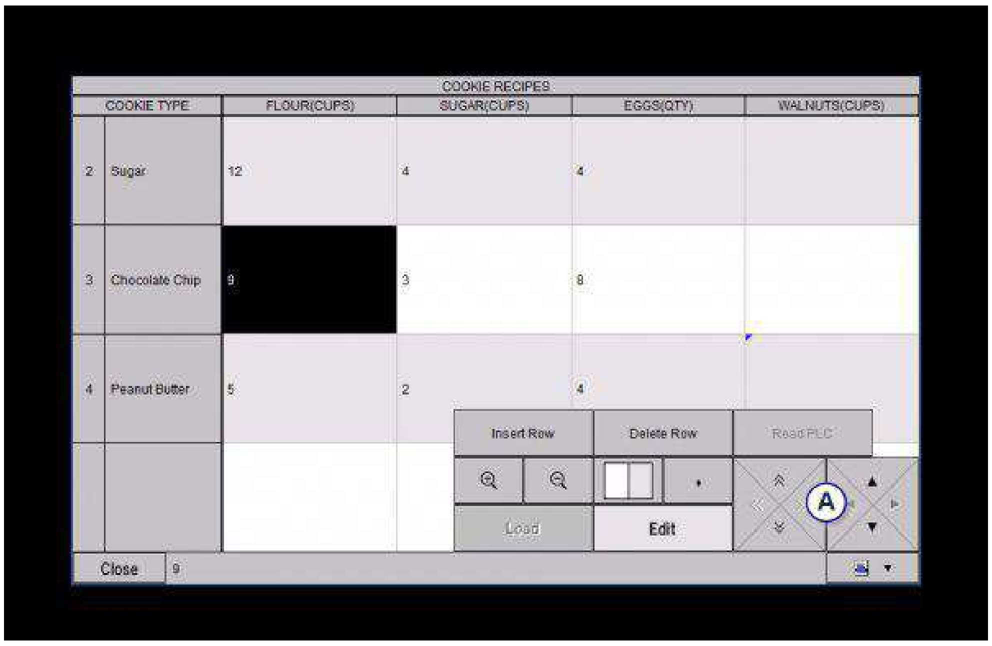
COOKIE RECIPES

COOKIE TYPE		FLOUR(CUPS)	SUGAR(CUPS)	EGGS(QTY)	WALNUTS(CUPS)
1	Oatmeal	10	5	10	
2	Sugar	12	4	4	
3	Chocolate Chip	9	3	8	
4	Peanut Butter	5	2	4	

Figure 5

In order for the operator to access the recipe functions, they will need to open the toolbox by touching the button in the lower right corner of the screen (Figure 5, item A). The toolbox (Figure 6) will allow the operator to scroll through the recipe list, edit tags, and load the selected recipe into the destination tags.

To scroll through the recipes use the arrow buttons located on the toolbox (Figure 6, item A). To change the value in a recipe, select Edit, then select the cell to be edited and enter a value using the pop up keypad. To load the recipe, select the recipe name, such as Oatmeal, and press the Load button.



COOKIE RECIPES

COOKIE TYPE		FLOUR(CUPS)	SUGAR(CUPS)	EGGS(QTY)	WALNUTS(CUPS)
2	Sugar	12	4	4	
3	Chocolate Chip	9	3	8	
4	Peanut Butter	5	2	4	

Figure 6

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

For further information on the Call Recipe object, reference the Help files included with the C-more programming software or download the latest Help file from: https://support.automationdirect.com/products/cmore.html

Learn More...Watch The Video!

New C-more Software Recipe Functions

http://n2adc.com/lhu0z

Data Logging and Trends

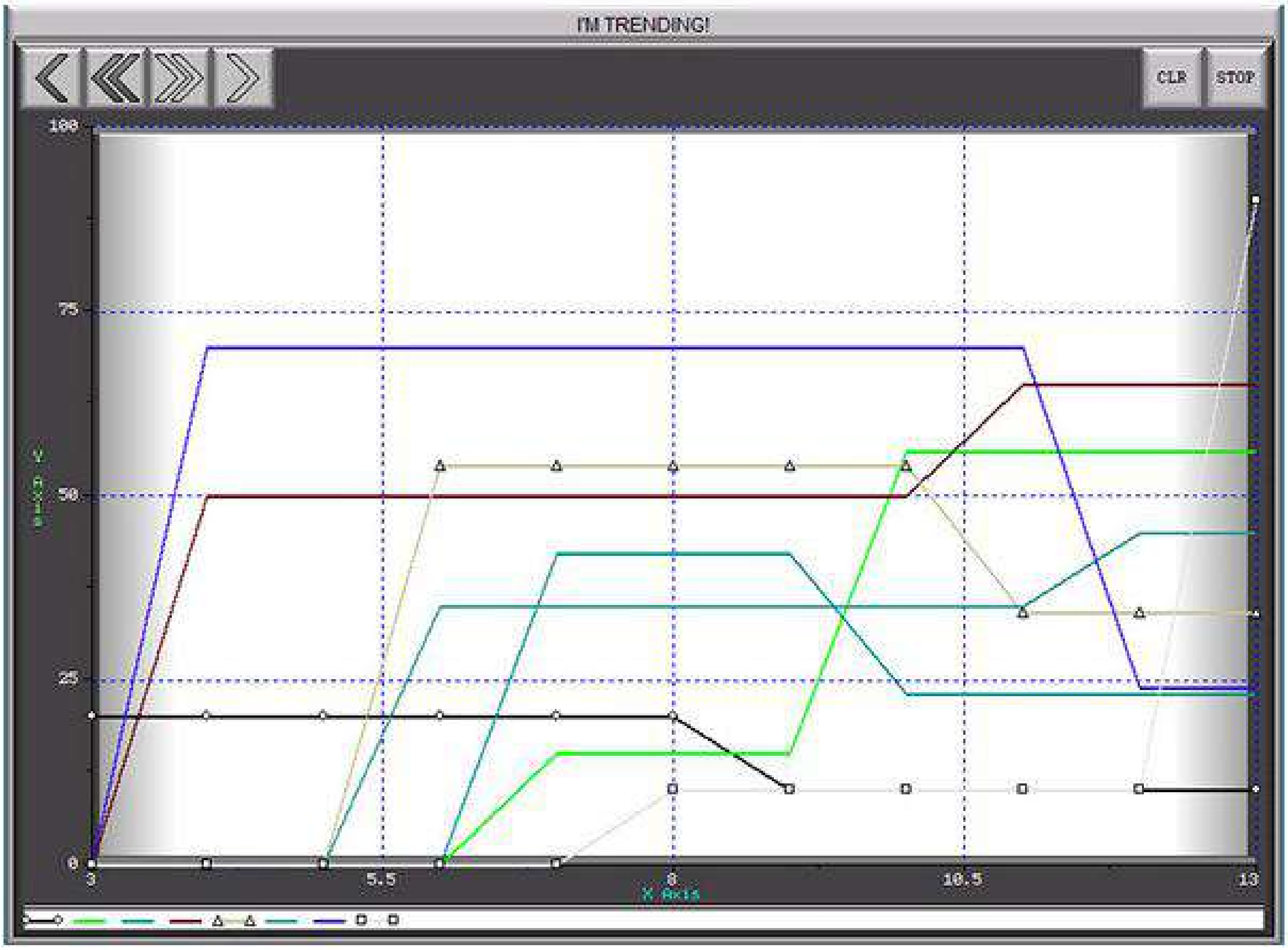
Operational data is extremely important and gives us a window into system performance, production throughput, and alerts us to potential problems. The C-more HMI data logging function allows you to store trend graphs which provide historical records of system events and actions. C-more will also record messages, alarms and screen captures all of which are very important especially when troubleshooting malfunctions.

What data is available?

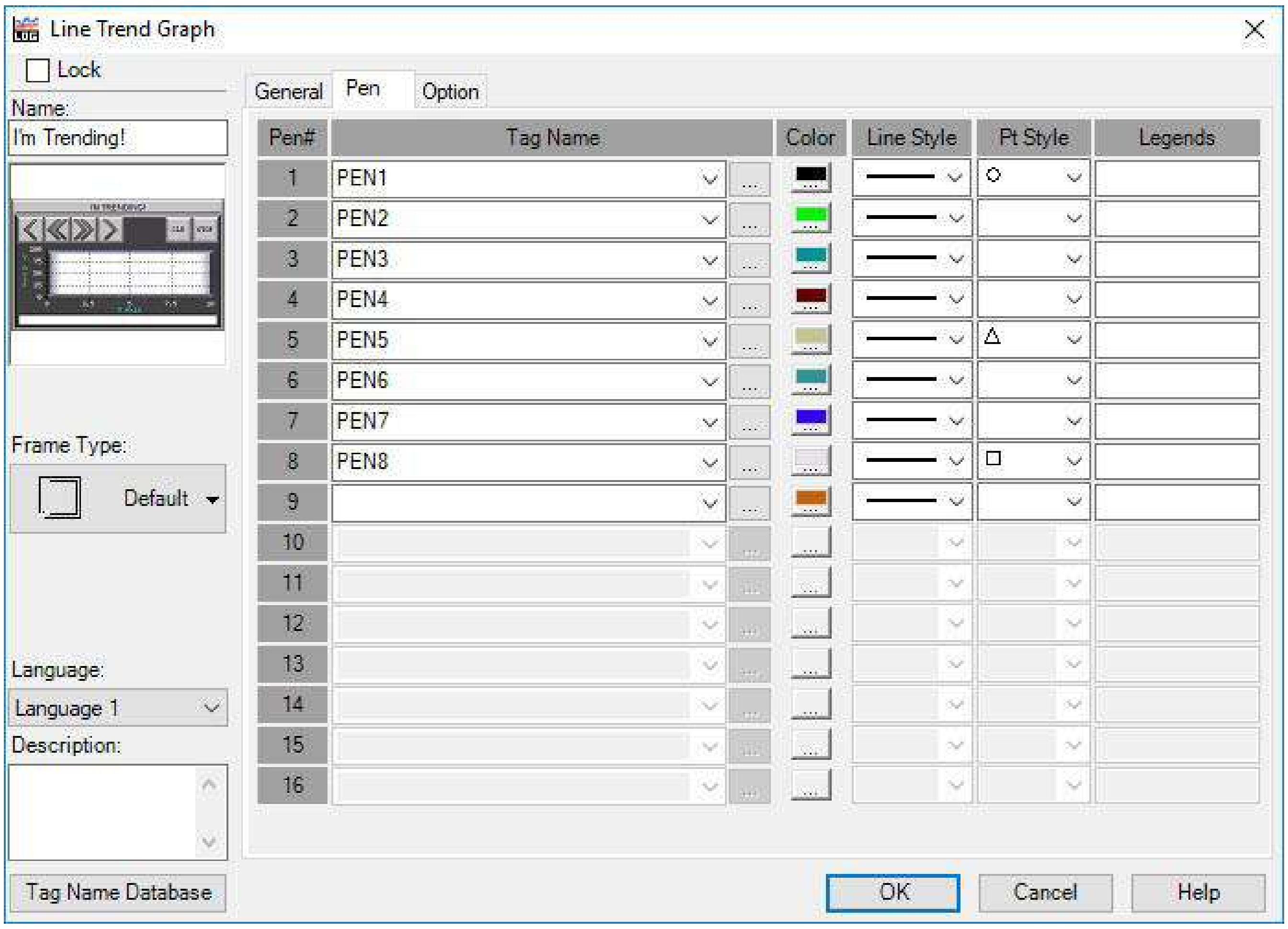
- Message Log Files - the latest Message log file created using the Lookup Text object and the Message Database
- Alarm Log Files - the latest log file produced in the Event Manager Alarm Action
- Line Trend Log Files - when selected, the latest Line Trend Log files for each Line Trend that has logging configured
- PID Faceplate Log Files- when selected, the latest PID Faceplate Line Trend Log files for each PID Faceplate Line Trend that has logging configured
- Screen Capture Images - the latest screen capture image recorded by the Event Manager Screen Capture Action

See what's trending

As mentioned above, data logging for the C-more panel is available through the Line Trend Graph and PID Faceplate Trend Graph Objects. The Line Trend Graph Object is designed to monitor an assigned tag name continuously. This object is very much like a digital version of a chart recorder used for industrial processes. This object can be configured to monitor up to 16 different tag names.



The 16 tag names are represented using pen lines and they are configured using the Pen Tab Configuration Window. Each pen can have its own color, line style, point style and legend.



Line Trend Graph

□ Lock

Name: I'm Trending!

Frame Type: Default

Language: Language 1

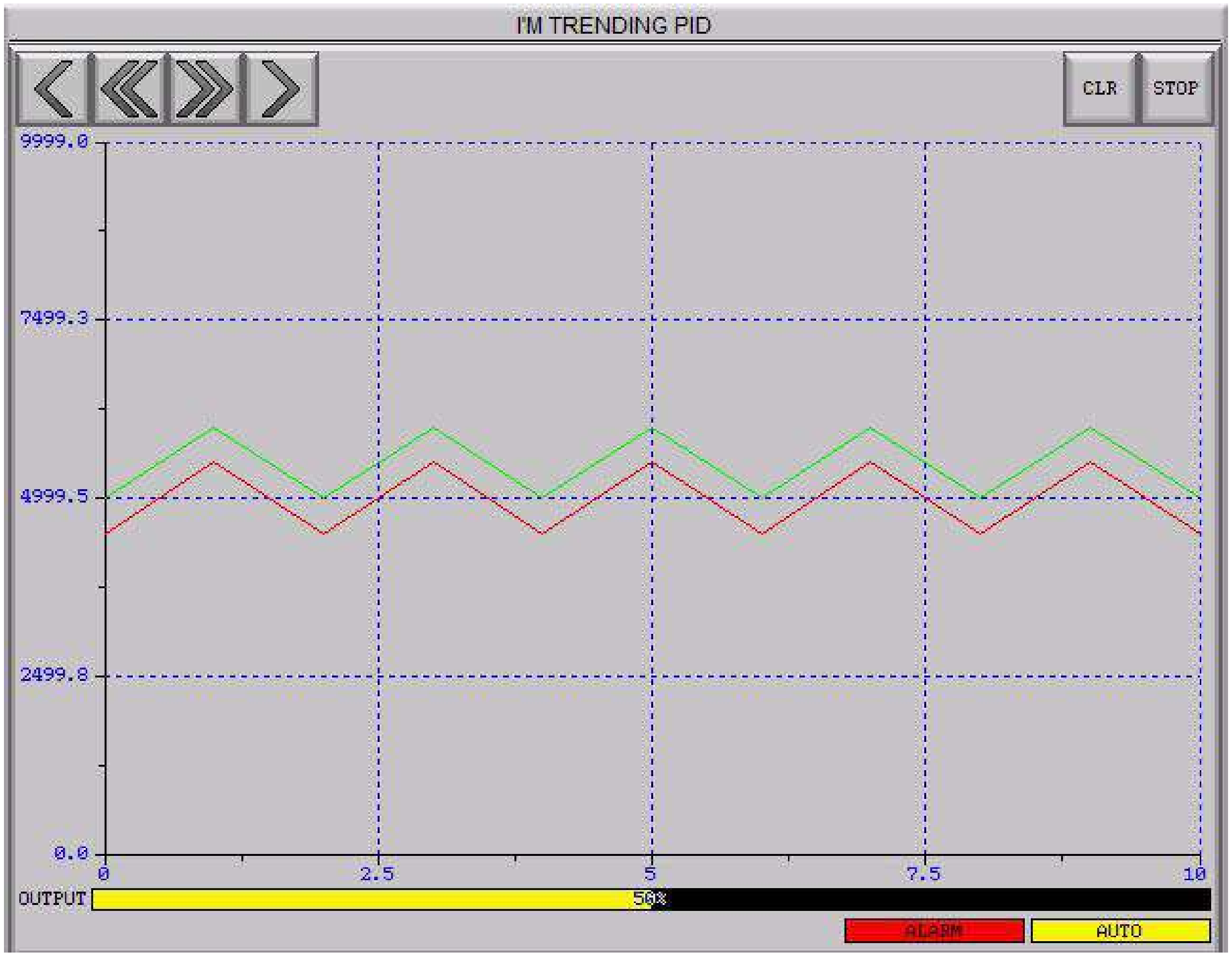
Description:

General | Pen | Option

Pen#	Tag Name	Color	Line Style	Pt Style	Legends
1	PEN1	■	———	○	
2	PEN2	🟩	———		
3	PEN3	■	———		
4	PEN4	🟥	———		
5	PEN5	■	———	△	
6	PEN6	■	———		
7	PEN7	🟦	———		
8	PEN8	□	———	□	
9		🟧	———		
10					
11					
12					
13					
14					
15					
16					

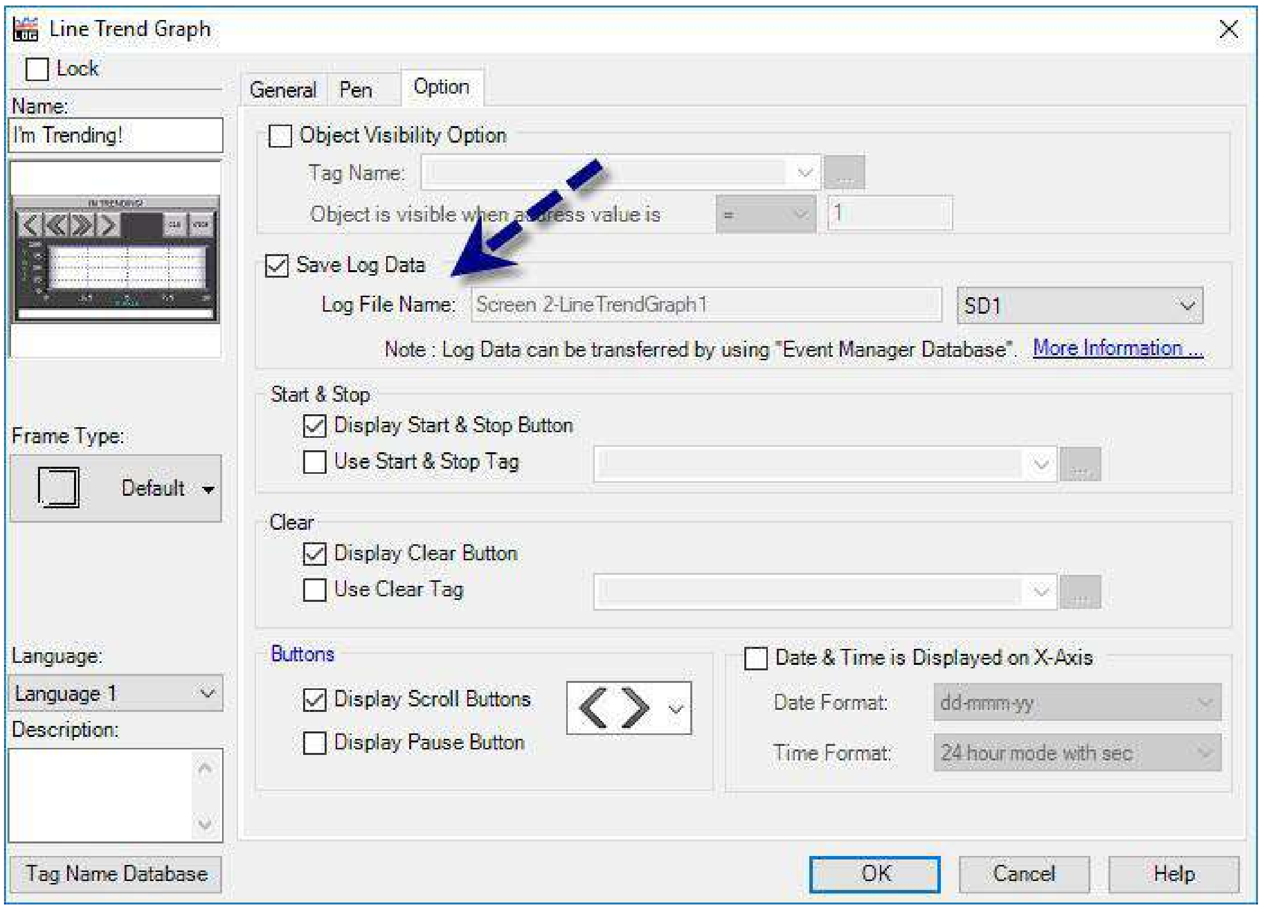
Tag Name Database | OK | Cancel | Help

The PID Faceplate Trend Graph Object is much like the Line Trend Graph Object except that it's used specifically to monitor control loop assigned tag names.



Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Both trend graphs are stored with the file name and in the location you choose in the Option tab of their configuration windows.



Line Trend Graph

□ Lock

Name: I'm Trending!

Frame Type: Default

Language: Language 1

Description:

□ Object Visibility Option

Tag Name:

Object is visible when address value is = 1

☑ Save Log Data

Log File Name: Screen 2-LineTrendGraph1 SD1

Note : Log Data can be transferred by using "Event Manager Database". More Information ...

Start & Stop

☑ Display Start & Stop Button

□ Use Start & Stop Tag

Clear

☑ Display Clear Button

□ Use Clear Tag

Buttons

☑ Display Scroll Buttons

□ Display Pause Button

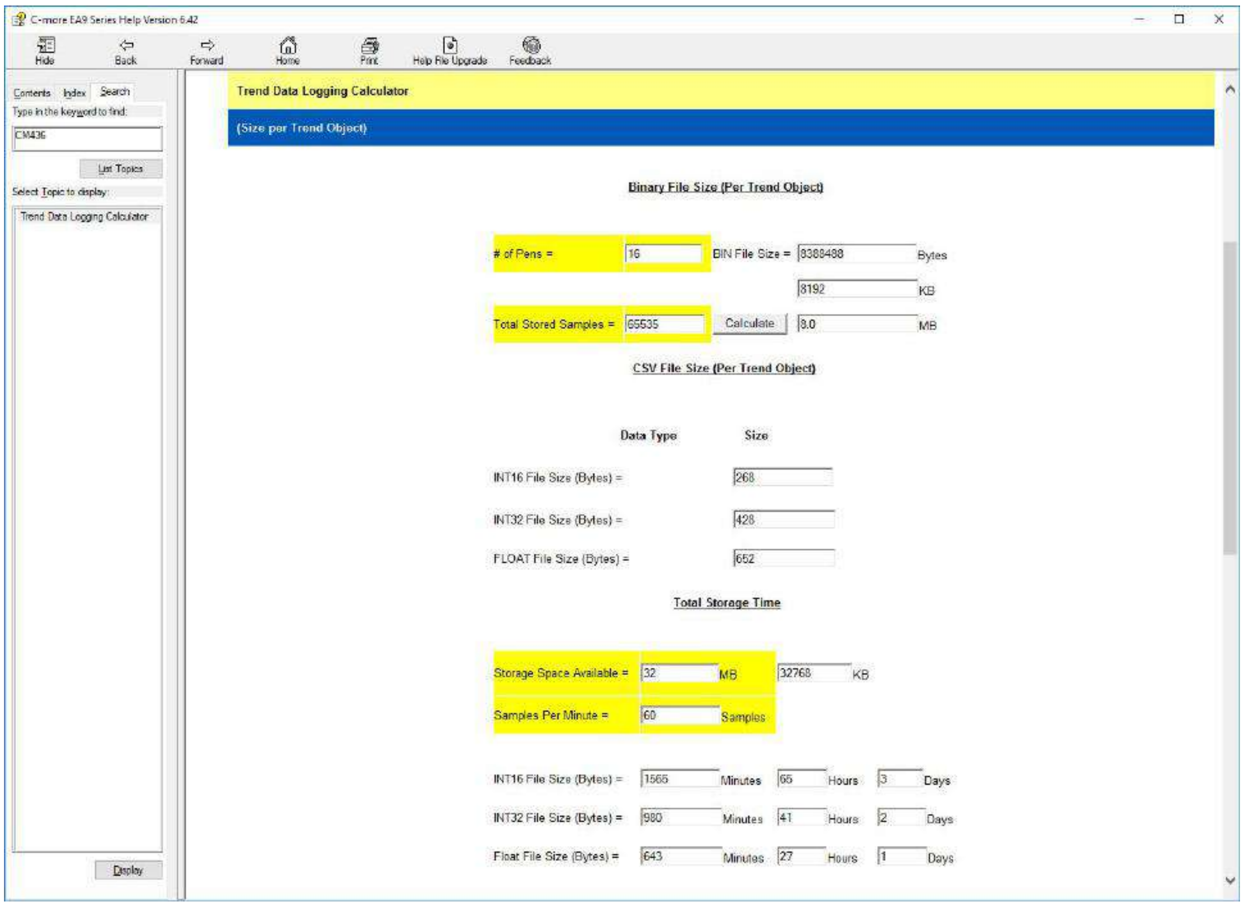
□ Date & Time is Displayed on X-Axis

Date Format: dd-mmm-yy

Time Format: 24 hour mode with sec

One thing to be mindful of when logging data with your C-more panel is the amount of memory available. The Panel Information window will provide you with information on the connected panel including the available memory in the MRAM and the installed USB and/or SD cards.

With that information and the convenient Trend Data Logging Calculator you can be sure you don't overrun your memory allocation. This calculator is available in the help file (topic CM436) and with it you can determine the needed memory size and the total storage time that will be recorded.



C-more EA9 Series Help Version 6.42

Type in the keyword to find: CM436

Select Topic to display: Trend Data Logging Calculator

Trend Data Logging Calculator

(Size per Trend Object)

Binary File Size (Per Trend Object)

# of Pens = 16 BIN File Size = 8388488 Bytes

8192 KB

Total Stored Samples = 65535 Calculate 8.0 MB

CSV File Size (Per Trend Object)

Data Type	Size
INT16 File Size (Bytes) =	268
INT32 File Size (Bytes) =	428
FLOAT File Size (Bytes) =	652

Total Storage Time

Storage Space Available = 32 MB 32768 KB

Samples Per Minute = 60 Samples

INT16 File Size (Bytes) =	1565	Minutes	65	Hours	3	Days
INT32 File Size (Bytes) =	980	Minutes	41	Hours	2	Days
Float File Size (Bytes) =	643	Minutes	27	Hours	1	Days

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

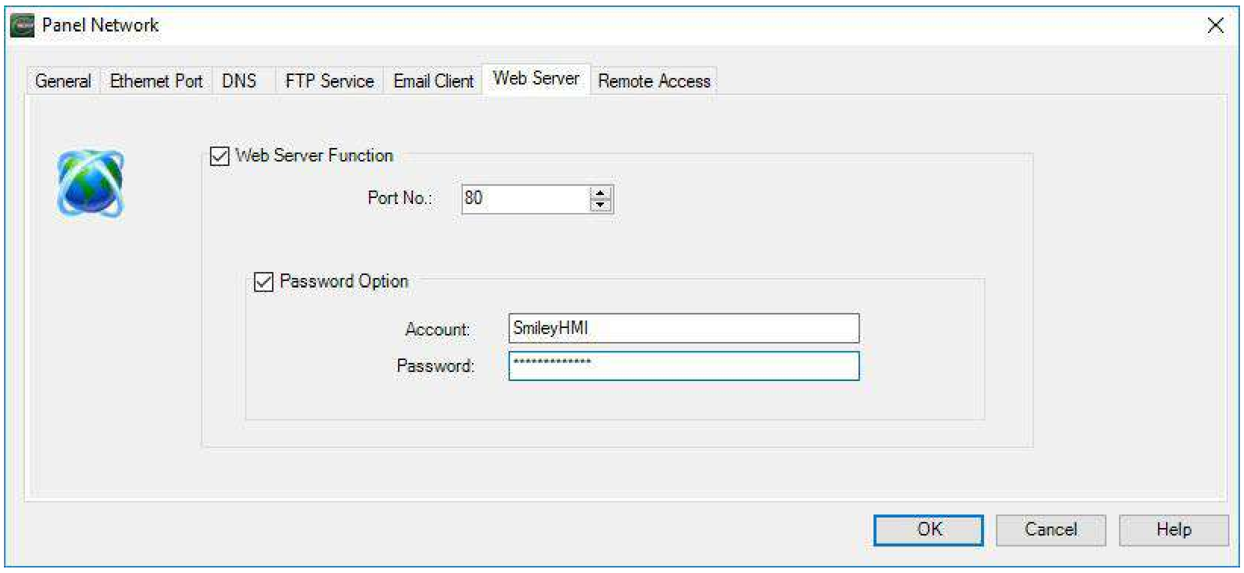
Seize the data!

Getting your C-more panel configured for data logging is a simple process but now the question becomes, how do you access the stored data? There are several ways of doing this.

1. Simply walk up to the machine or system you are monitoring, remove the storage device and view the data files with your PC
2. Connect to the panel over your Ethernet network and view the stored data using the embedded Web server
3. The C-more HMI can transfer the data files to your PC using the FTP data transfer function
4. The C-more HMI could also email the data files directly to you or any other need-to-know personnel

Web Server

Configuring your panel for Web Server access is a cinch. Simply open the Panel Network dialog box from the Setup menu. Under the Web Server tab select the Web Server Function checkbox. Change the port number if needed and assign a password to keep your panel secure.



☑ Web Server Function

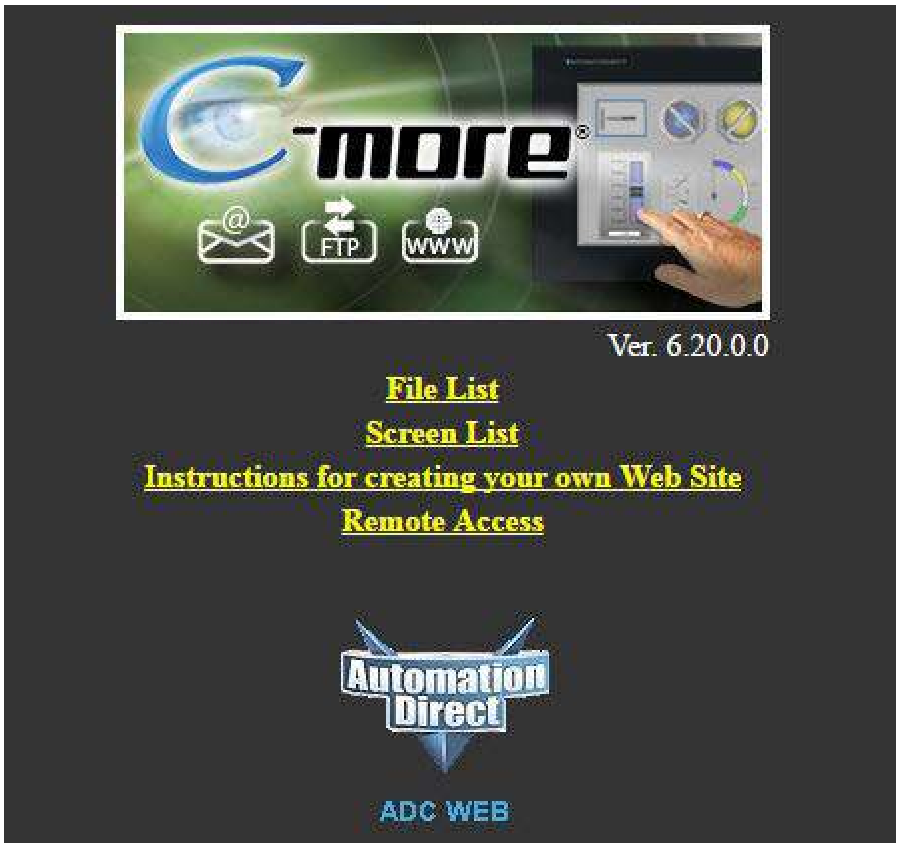
Port No.: 80

☑ Password Option

Account: SmileyHMI

Password: *************

After downloading the changes, open a browser window and enter the IP address of the panel in the Address field. You will then access your files and screens.



Ver. 6.20.0.0

File List

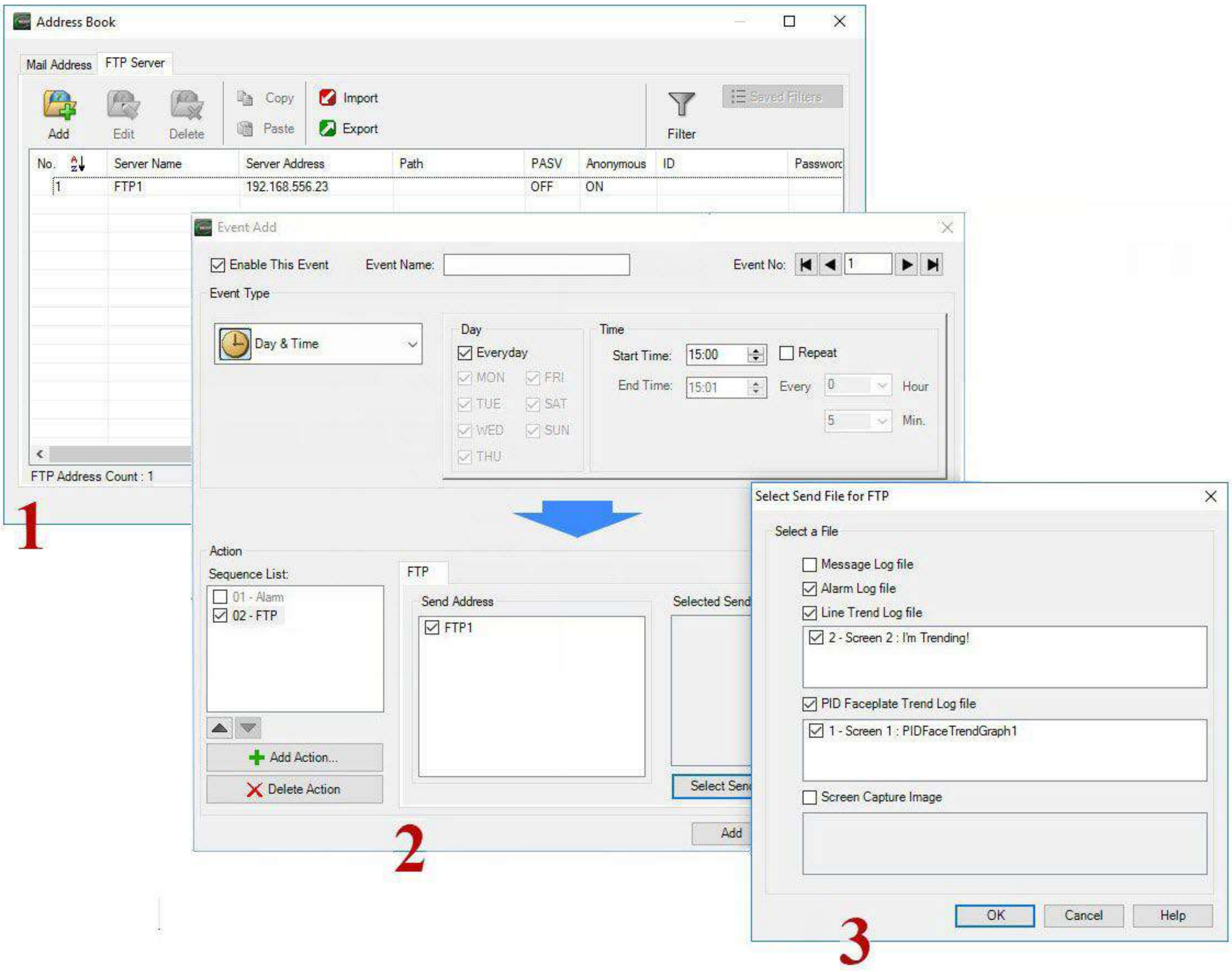
Screen List

Instructions for creating your own Web Site

Remote Access

FTP Client Data Transfer

The FTP client (File Transfer Protocol) allows transfer of data files between C-more panels and any connected FTP compatible device. Your HMI can collect data and pass it to the PC on your desk, or to your corporate headquarters.



No.	Server Name	Server Address	Path	PASV	Anonymous	ID	Password
1	FTP1	192.168.556.23		OFF	ON		

☑ Enable This Event Event Name: Event No: 1

Event Type: Day & Time

Day: ☑ Everyday ☑ MON ☑ TUE ☑ WED ☑ THU ☑ FRI ☑ SAT ☑ SUN

Time: Start Time: 15:00 End Time: 15:01 ☐ Repeat Every 0 Hour 5 Min.

Sequence List: ☐ 01 - Alarm ☑ 02 - FTP

FTP: Send Address ☑ FTP1 Selected Send

☐ Message Log file
☑ Alarm Log file
☑ Line Trend Log file
☑ 2 - Screen 2 : I'm Trending!
☑ PID Faceplate Trend Log file
☑ 1 - Screen 1 : PIDFaceTrendGraph1
☐ Screen Capture Image

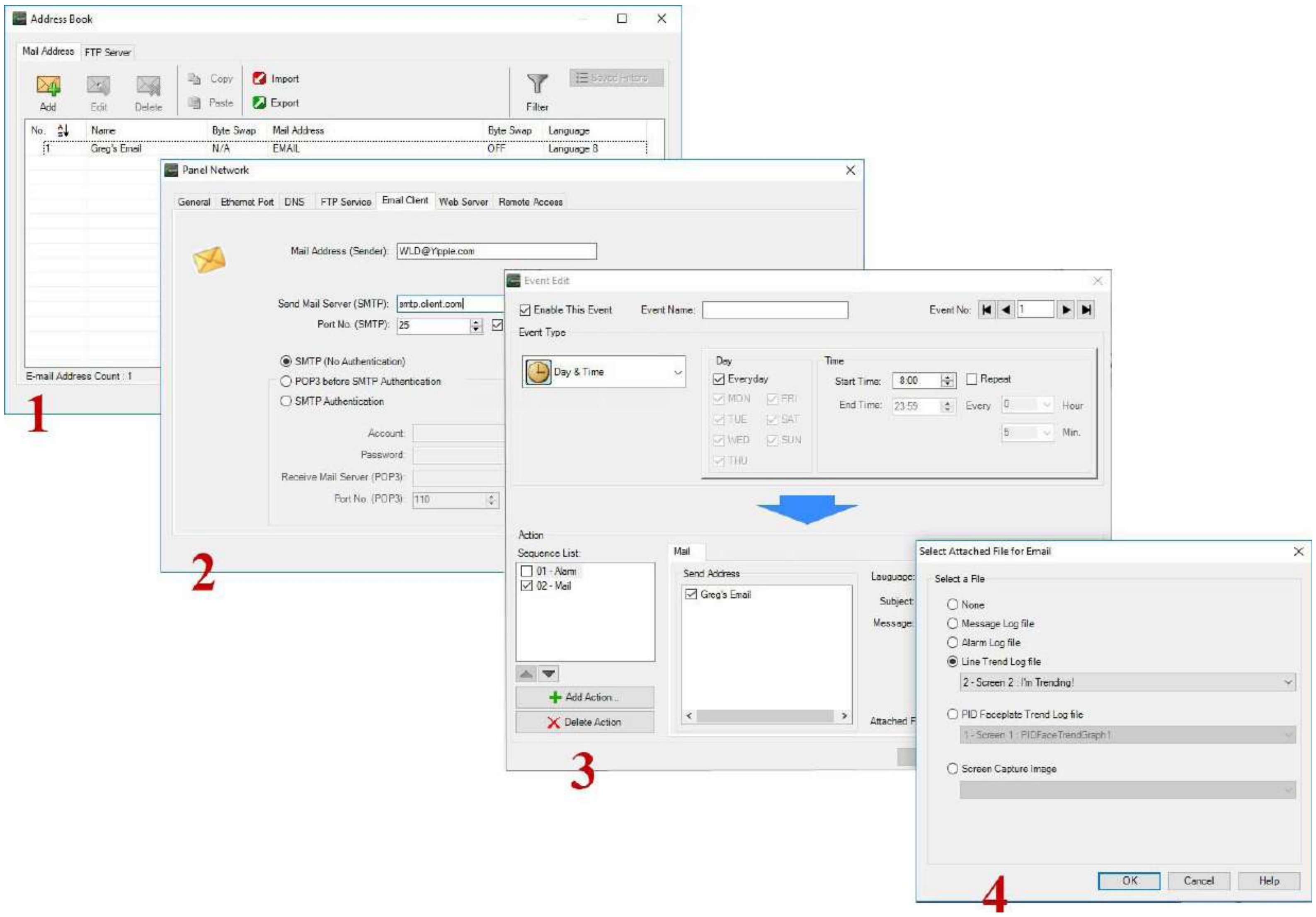
Assign the FTP server address in the Address Book, configure the FTP transfer as an action in the Event Manager based upon a day/time, tag event or screen change, then choose which files to send.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

E-mail

The E-mail Client feature allows the touch panel to be set up to send e-mail messages from the panel with embedded data, date and time stamps and attached files containing logged data, trend graphs and screen captures. This feature requires an SMTP (Simple Mail Transfer Protocol) email server and optionally a DHCP (Dynamic Host Configuration Protocol) server and/or DNS (Domain Name System).

Similar to the FTP server option, assign the e-mail address in the Address Book (either a static address or tag name), enable the E-mail Client in the Panel Network window, configure the e-mail, with data log attachment, as an action in the Event Manager based upon a day/time, tag event or screen change, then choose which file to send.



No.	Name	Byte Swap	Mail Address	Byte Swap	Language
1	Greg's Email	N/A	EMAIL	OFF	Language 8

E-mail Address Count : 1

Mail Address (Sender): WLD@Yippie.com
Send Mail Server (SMTP): smtp.client.com
Port No. (SMTP): 25 ☑
● SMTP (No Authentication)
○ POP3 before SMTP Authentication
○ SMTP Authentication
Account:
Password:
Receive Mail Server (POP3):
Port No. (POP3): 110

☑ Enable This Event Event Name: Event No: 1
Event Type: Day & Time
Day: ☑ Everyday ☑ MON ☑ TUE ☑ WED ☑ THU ☑ FRI ☑ SAT ☑ SUN
Time: Start Time: 8:00 □ Repeat
End Time: 23:59 Every 0 Hour 5 Min.

Sequence List:
□ 01 - Alarm
☑ 02 - Mail
Send Address: ☑ Greg's Email
Lauguage:
Subject:
Message:
Attached F

○ None
○ Message Log file
○ Alarm Log file
● Line Trend Log file
2 - Screen 2 : I'm Trending!
○ PID Faceplate Trend Log file
1 - Screen 1 : PIDFaceTrendGraph1
○ Screen Capture Image

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Data in Real Time

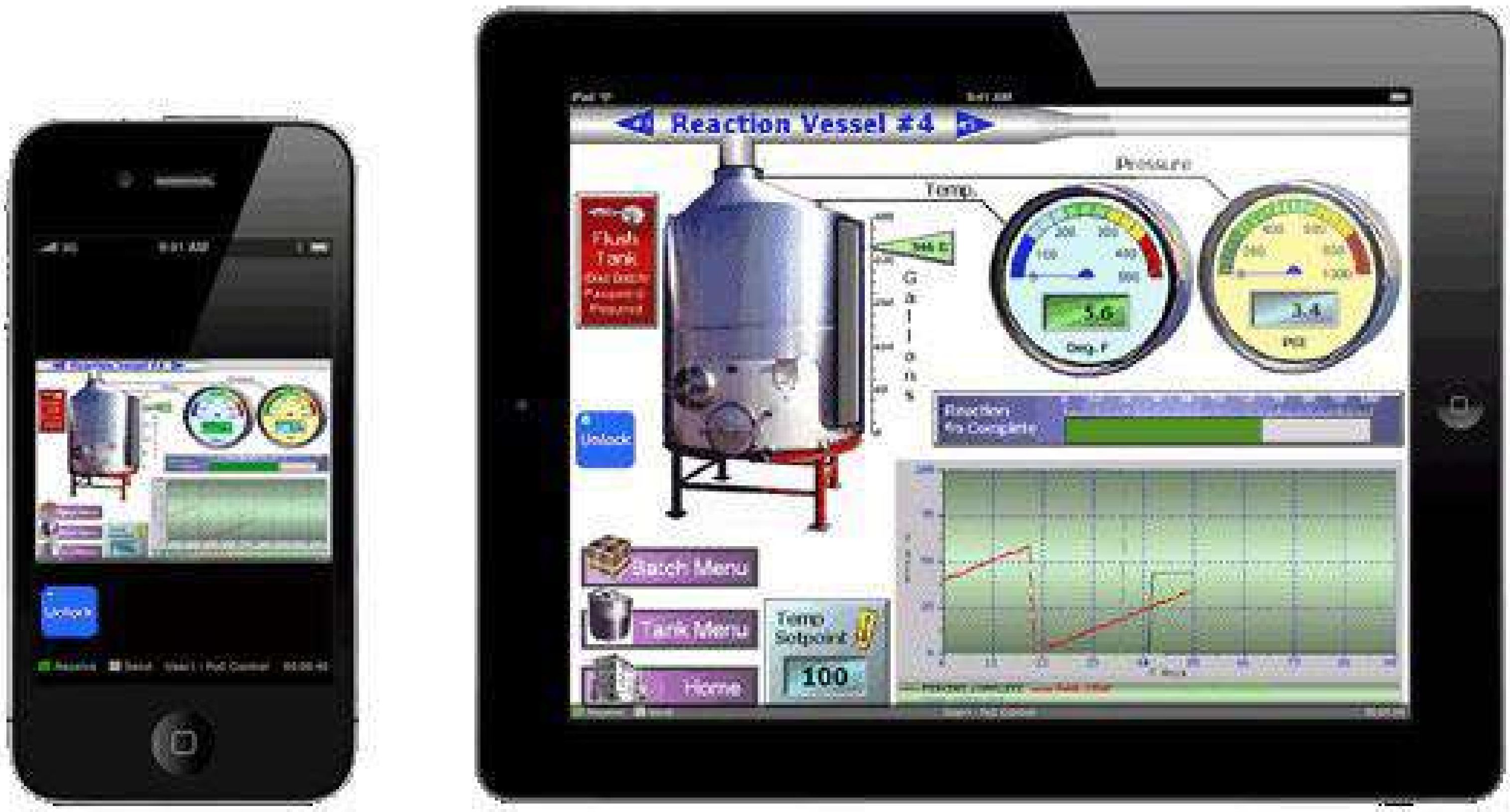
If you wish to view the data as it's being collected, you can also remotely view live data using the Remote Access feature and/or the C-more Remote HMI Mobile App.

Remote Access

Remote Access takes the Web Server functionality one step further and provides real-time HMI screens allowing the remote user to control, fully operate and monitor the local HMI system as if on site. Remote access is also configured within the Panel Network dialog and the Web Server option must be enabled to download the application.

C-more Mobile App

The C-more Remote HMI App allows authorized users to connect their mobile device to the C-more panel from a Wi-Fi or cellular connection. The remote user can fully operate and/or monitor a remote C-more panel. The app is extremely convenient for users who need to access operational data but are away from their PC or at a remote site with limited connectivity.



See Remote Monitoring Chapter for more info

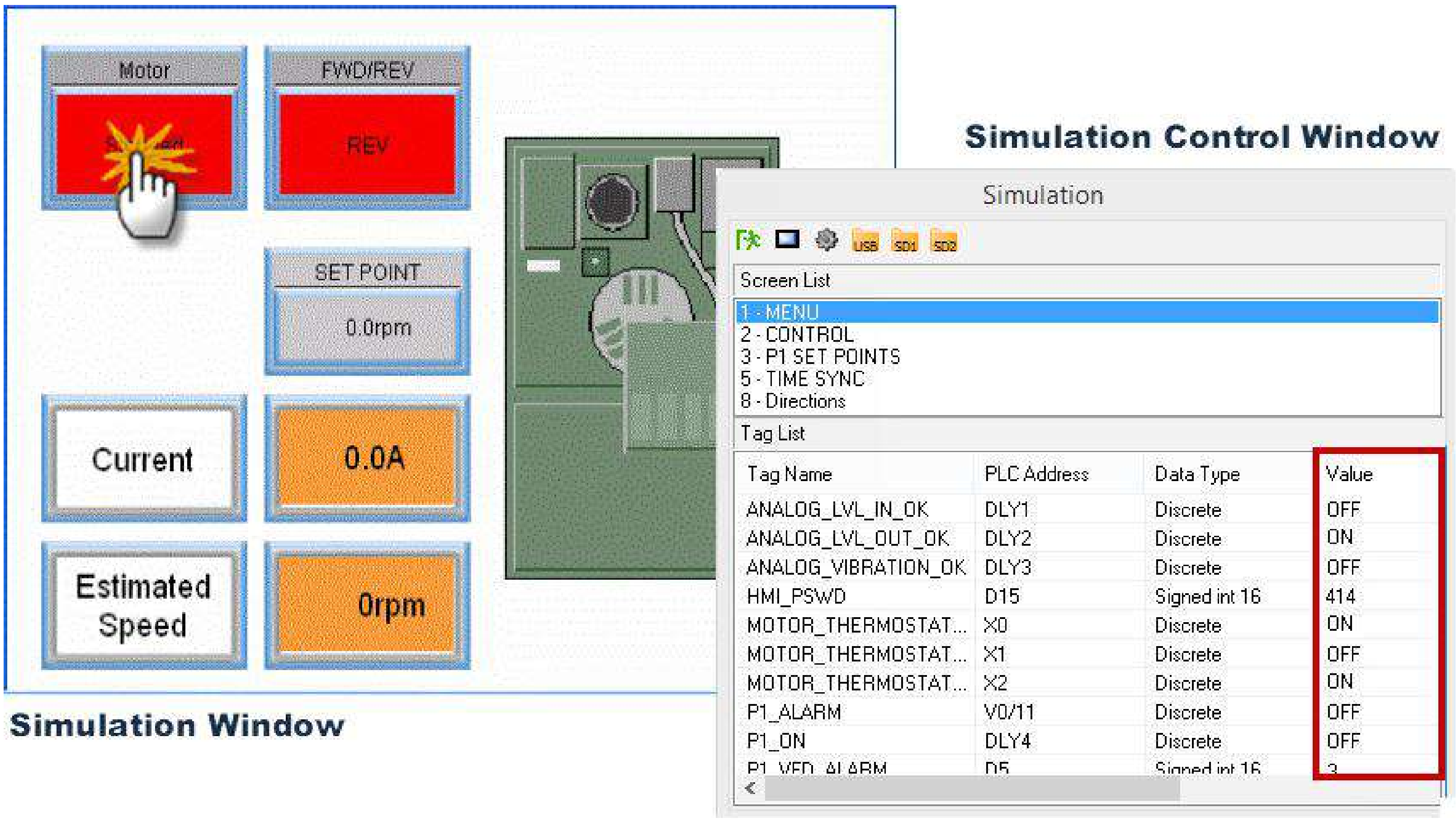
Note: For those who prefer to use the Remote Access, Web Server, and/or Mobile App features of the C-more panel instead of the local display, the C-more Headless HMI is the way to go. The headless HMI provides all the same functionality as the touch screen HMI without paying for a local display you won't use. A local HDMI display or touch screen monitor can be added if needed in the future.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Project Simulation

Modern HMIs provide an extensive pallet of graphical elements to be used when designing screens. The number of objects and options available is immense and verifying these objects are functioning as designed can be a daunting task if not done continually during the design process. Imagine configuring an analog meter in your project and deciding to use that same meter in 6 locations on 10 process screens. You later realize, during deployment of the new screens, that the original meter's visibility was not configured properly and therefore all 60 are wrong. Not fun! Luckily, the C-more HMI provides a way to quickly verify each object's functionality through project simulation. In our example, simulating the original analog meter to verify it's correct before duplicating it would have prevented a lot of rework.

The C-more simulation feature allows the user to simulate a project during development before it's ever transferred to a panel. When simulating a project, a Simulation Window is displayed allowing all of the objects in the project to be tested. Testing is done by either clicking an object on the simulation window with a mouse or by entering a value in the Value column of the Tag List section in the Simulation Control Window seen below.



Simulation Control Window

Screen List

1 - MENU
2 - CONTROL
3 - P1 SET POINTS
5 - TIME SYNC
8 - Directions

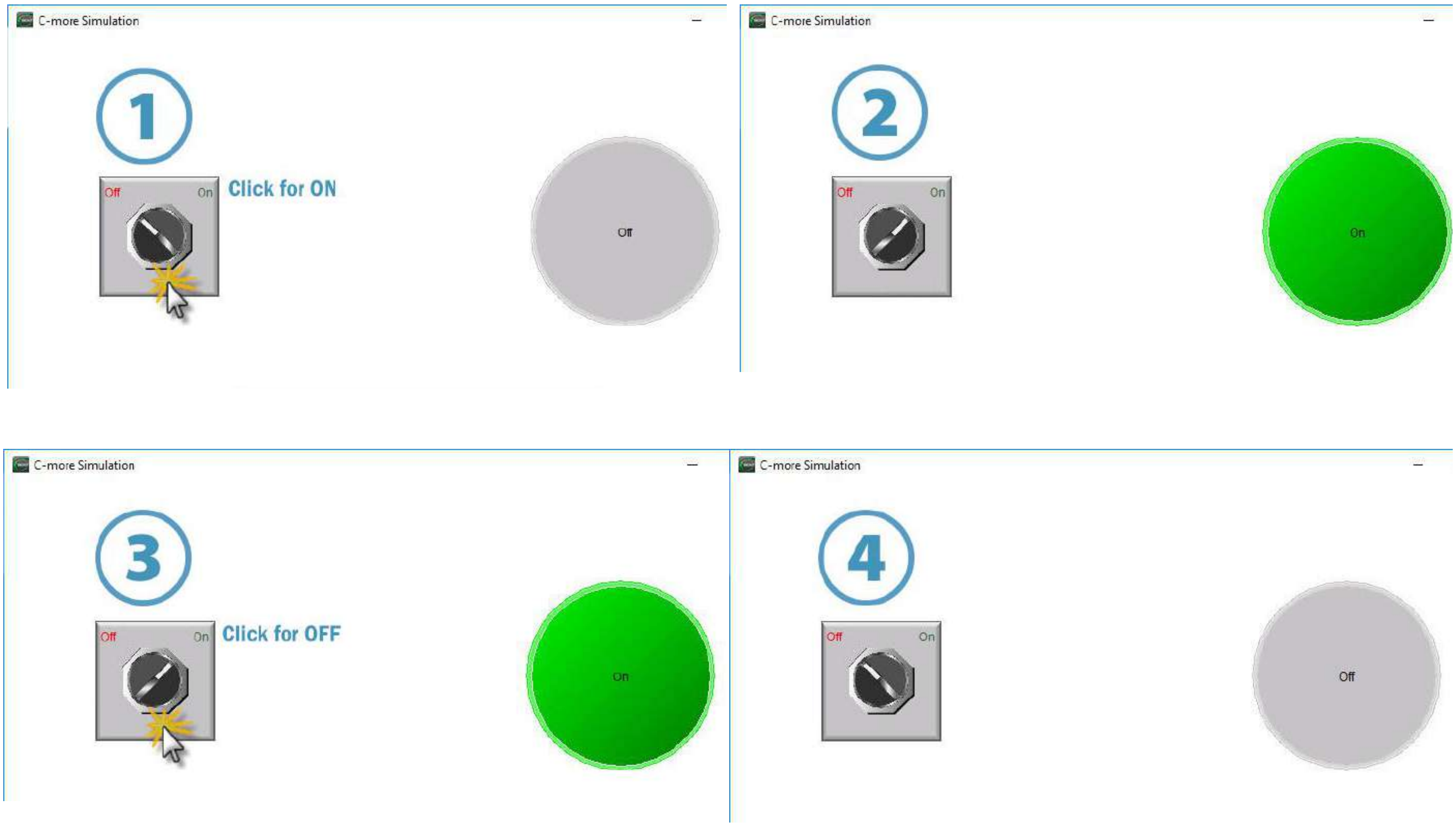
Tag List

Tag Name	PLC Address	Data Type	Value
ANALOG_LVL_IN_OK	DLY1	Discrete	OFF
ANALOG_LVL_OUT_OK	DLY2	Discrete	ON
ANALOG_VIBRATION_OK	DLY3	Discrete	OFF
HMI_PSWD	D15	Signed int 16	414
MOTOR_THERMOSTAT...	X0	Discrete	ON
MOTOR_THERMOSTAT...	X1	Discrete	OFF
MOTOR_THERMOSTAT...	X2	Discrete	ON
P1_ALARM	V0/11	Discrete	OFF
P1_ON	DLY4	Discrete	OFF
P1_VFD_ALARM	D5	Signed int 16	3

Simulation Window

C-more Simulation Window

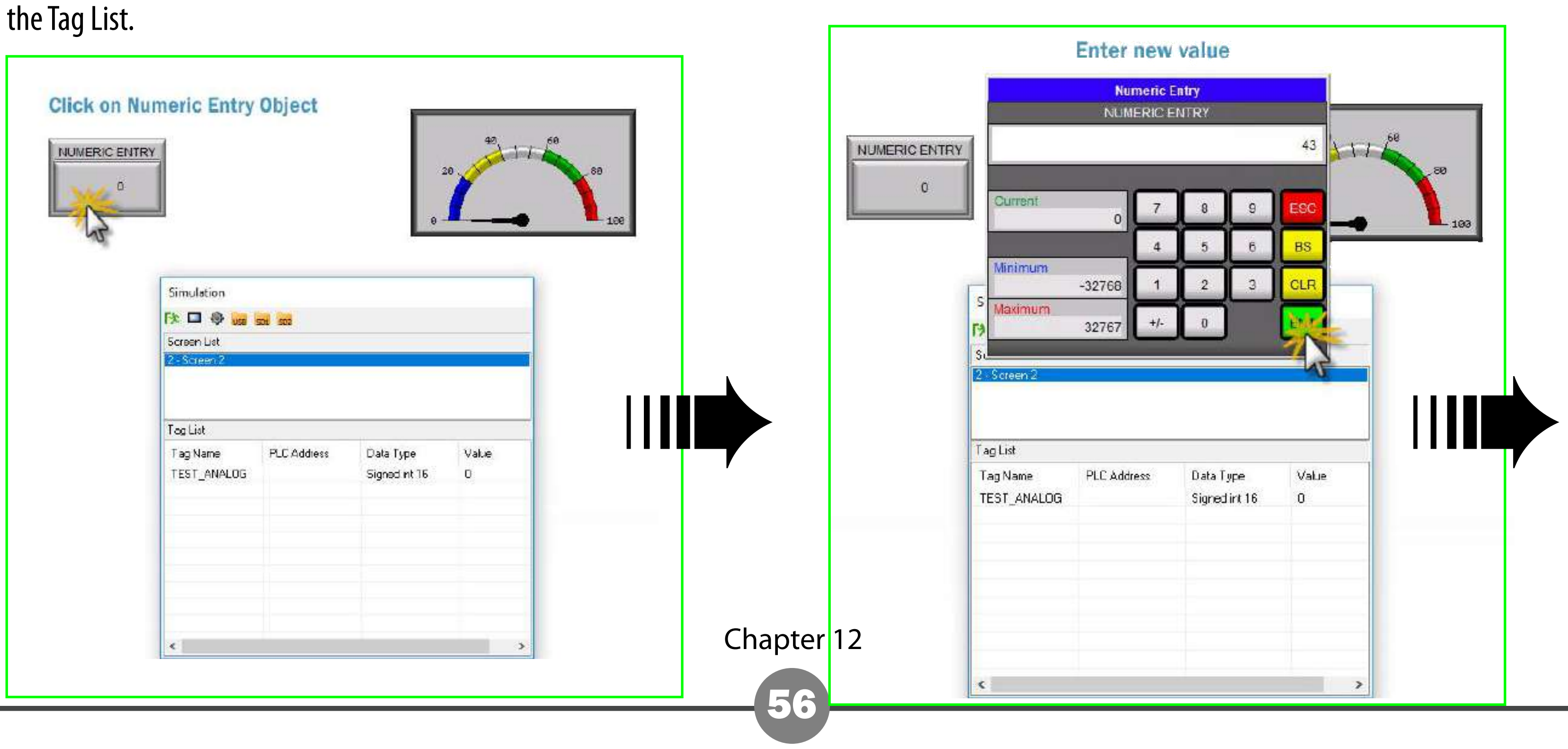
The C-more Simulation window displays the project screen as it will be seen when live on a panel. Discrete objects can be tested by clicking on the object. Numeric entry objects can be tested by clicking on the object and entering a value with the keypad. Meters, graphs, numeric displays and similar objects that are linked to numeric entry objects, will change along with the new entry.



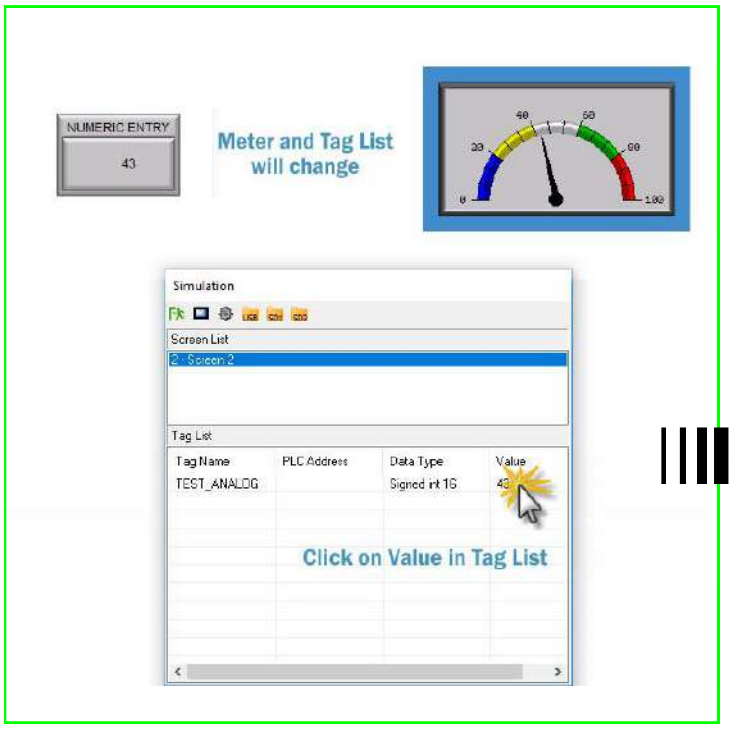
Simulation Control Window

The Simulation Control window provides a project Screen List to select the screen you wish to simulate and allows for direct control of a tag's value for testing meters, graphs, numeric displays and similar objects. The Tag List includes a Tag Name, PLC Address and Data Type for each object on the selected screen and objects are tested by entering a value in the Value field.

The example below demonstrates the two ways to simulate an analog meter object using a linked entry object or directly through the Tag List.



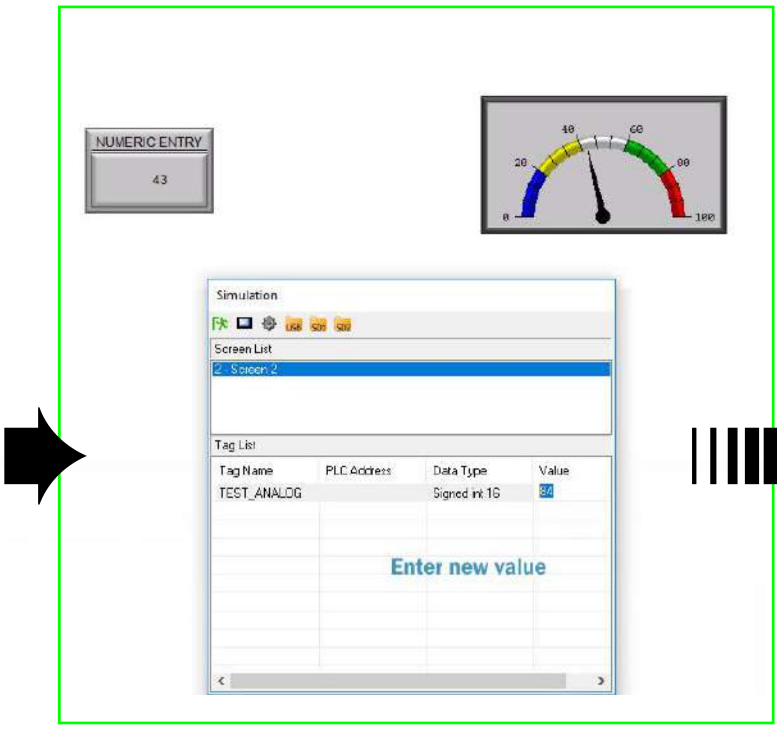
Tag Name	PLC Address	Data Type	Value
TEST_ANALOG		Signed int 16	0



Meter and Tag List will change

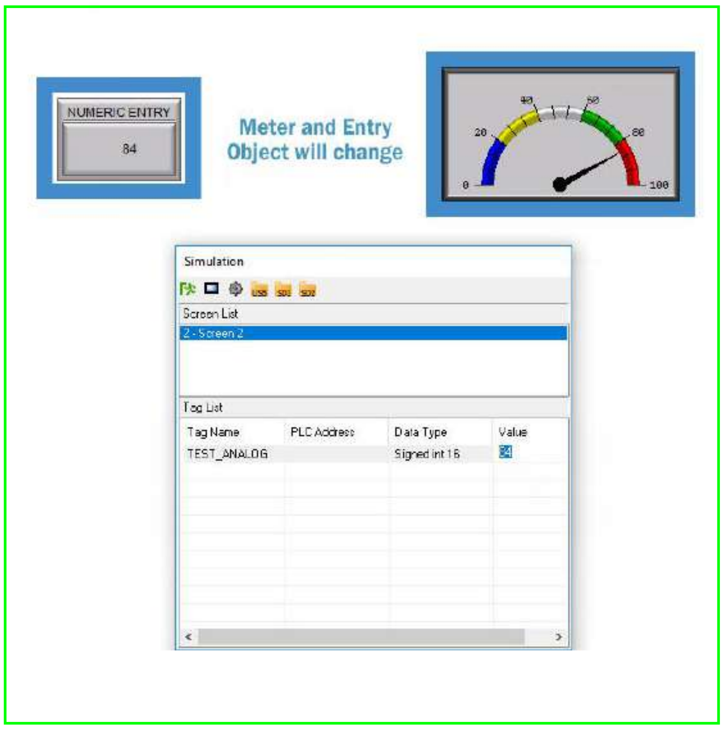
Tag Name	PLC Address	Data Type	Value
TEST_ANALOG		Signed int 16	43

Click on Value in Tag List



Tag Name	PLC Address	Data Type	Value
TEST_ANALOG		Signed int 16	84

Enter new value



Meter and Entry Object will change

Tag Name	PLC Address	Data Type	Value
TEST_ANALOG		Signed int 16	84

Check out how the simulator can be a huge help in this short video

Learn More...Watch The Video!

C-more HMI : Simulate your HMI Project

C-more HMI: Simulate your HMI Project
Watch later
Share
C-more HMI
C-more
TECHTIPS
Simulate Your HMI Project

http://n2adc.com/u7lbq

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Tutorial: Simple AC Drive Control Interface

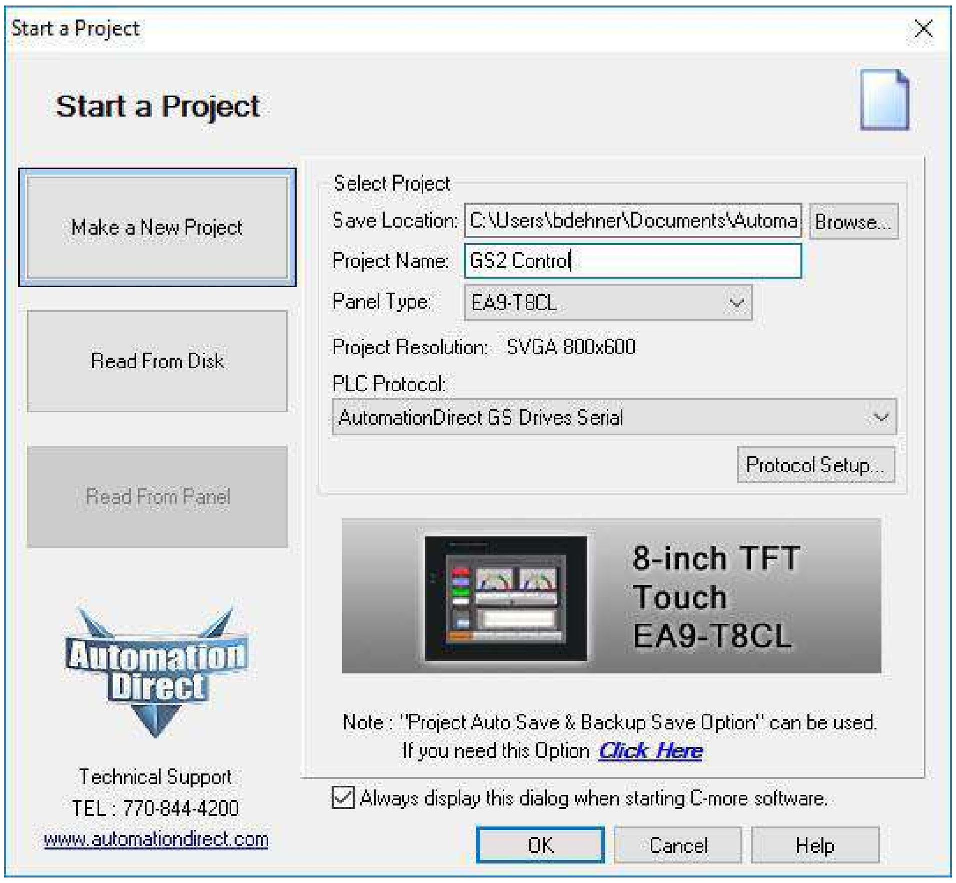
This tutorial demonstrates how to create a simple AC drive control interface. The purpose of this exercise is to show you the steps involved when designing an HMI screen. Of course, we will use the C-more programming software but many of these steps are similar to other HMI packages. We do not intend to imply suitability of this project for your application.

In this example, a C-more EA9-T8CL panel provides the control interface to a GS2 AC drive. The panel communicates with the drive via Modbus RTU protocol.

For this tutorial, the HMI will have:

- 2 buttons - to control the start/stop of the AC drive and select the direction of rotation
- 1 setpoint button - to control speed
- 2 display windows - one for estimated speed and one for current reading
- 1 bitmap image of the process being controlled

Step 1: Start a Project



Start a Project

Start a Project

Make a New Project

Read From Disk

Read From Panel

Select Project

Save Location: C:\Users\bdehner\Documents\Automa Browse...

Project Name: GS2 Control

Panel Type: EA9-T8CL

Project Resolution: SVGA 800x600

PLC Protocol:

AutomationDirect GS Drives Serial

Protocol Setup...

Note : "Project Auto Save & Backup Save Option" can be used. If you need this Option Click Here

Technical Support
TEL : 770-844-4200
www.automationdirect.com

☑ Always display this dialog when starting C-more software.

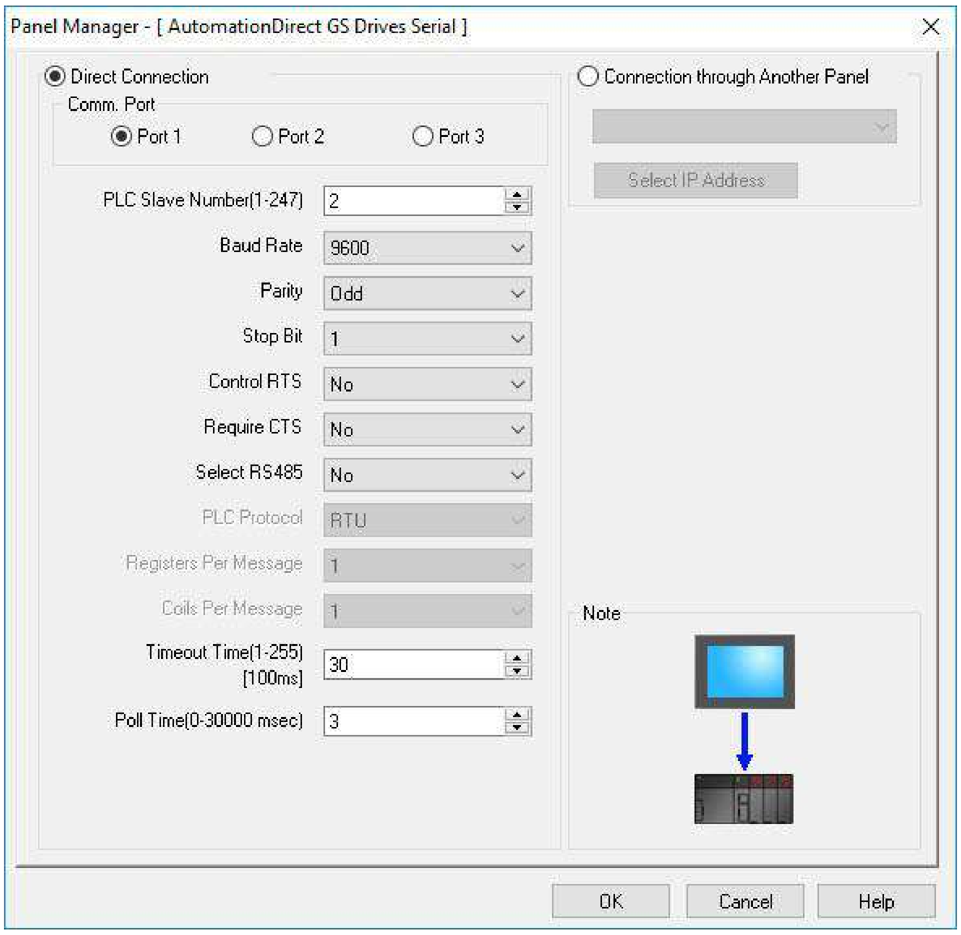
OK Cancel Help

1. Open the C-more Programming Software.
2. From the Start a Project window be sure the Make a New Project option is selected.
3. Click the Browse button in the Save Location field if you wish to change the location of the saved file. The default location is the Project folder of the C-more application.
4. Give your project a name in the Project Name field. I went with "GS2 Control".
5. Select the type of HMI you are using for this application. Click the drop-down arrow in the Panel Type field and select the EA9-T8CL.
6. Select the proper PLC protocol from the drop-down list in the PLC protocol field. Since we are using a GS series drive, I selected the AutomationDirect GS Drives Serial option.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

7. Click on the Protocol Setup button to configure the protocol specific to your application.

   a. In the Direct Connection configuration panel, the C-more panel will be configured as PLC Slave Number = 2, Baud Rate = 9600, Parity = ODD, Stop Bit = 1, Select RS485 = No, Timeout Time = 30ms and Poll Time = 3ms. The remaining parameters will use the default values. Refer to the Protocol Setup topic in the help file for detailed information on each of the protocols available for C-more panel communication.



Panel Manager - [AutomationDirect GS Drives Serial]

● Direct Connection
Comm. Port
● Port 1 ○ Port 2 ○ Port 3

○ Connection through Another Panel
Select IP Address

PLC Slave Number(1-247): 2
Baud Rate: 9600
Parity: Odd
Stop Bit: 1
Control RTS: No
Require CTS: No
Select RS485: No
PLC Protocol: RTU
Registers Per Message: 1
Coils Per Message: 1
Timeout Time(1-255) [100ms]: 30
Poll Time(0-30000 msec): 3

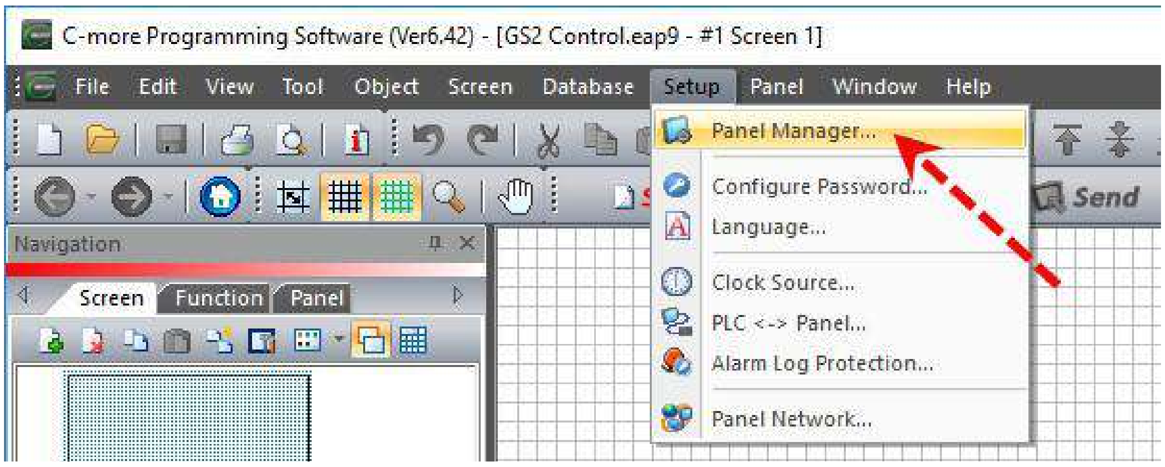
Note

OK Cancel Help

In our example, these values match the parameters that would have been programmed for the GS2 group 9:
P9.00 => 2 (slave #)
P9.01 => 01 (transmission speed)
P9.02 => 05 (MODBUS RTU, 8 data bits, Odd parity, 1 stop bit)

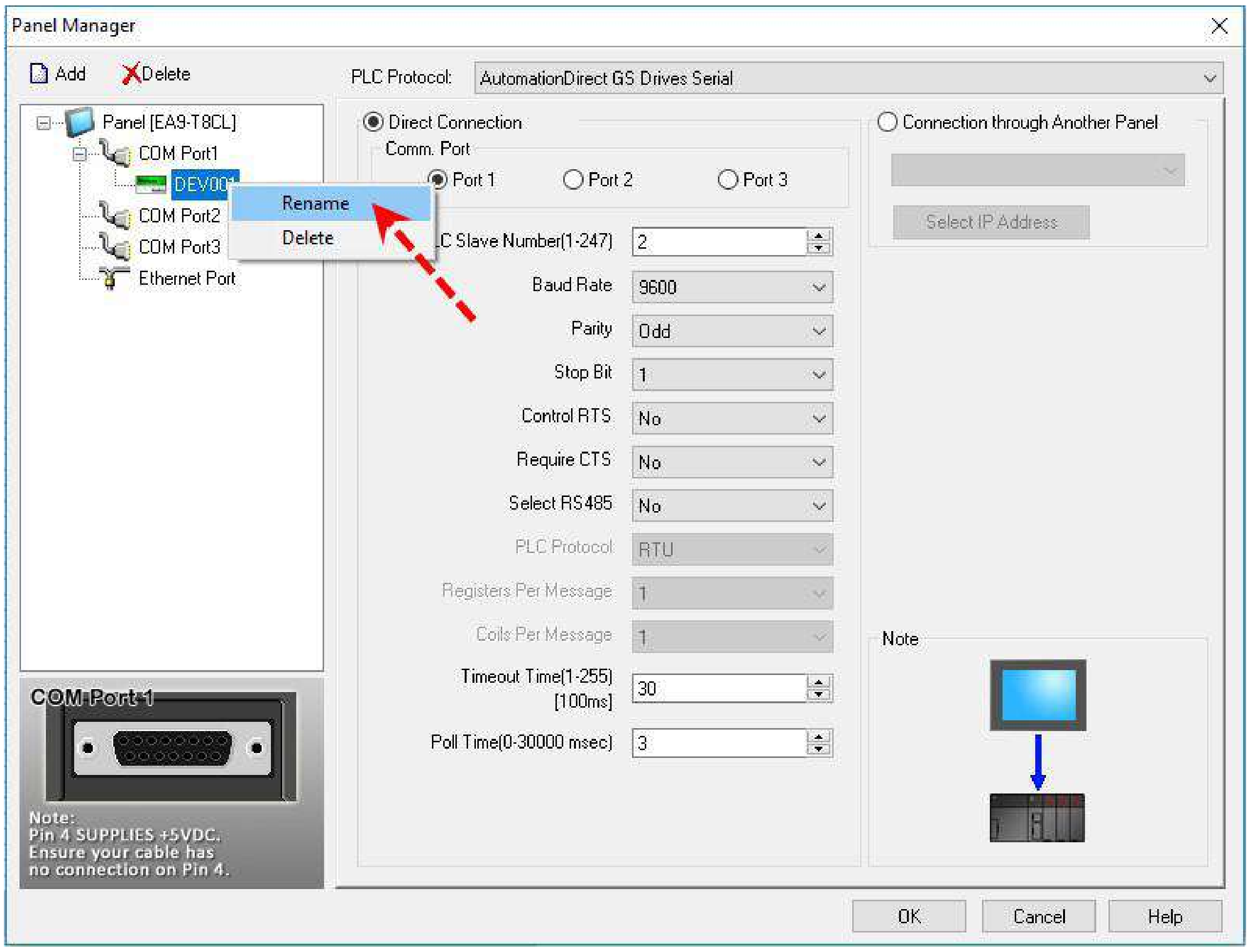
8. After verifying entries and selections in the Direct Connection panel, click OK in the Panel Manager protocol setup window, then click OK in the Start a Project window.

9. Access the Panel Manager window by clicking on Setup on the main window toolbar and selecting Panel Manager from the drop-down menu.



C-more Programming Software (Ver6.42) - [GS2 Control.eap9 - #1 Screen 1]

10. Right click on DEV001 and select Rename, then replace "DEV001" with "GS2" then click OK.



Panel Manager

PLC Protocol: AutomationDirect GS Drives Serial

● Direct Connection
Comm. Port: ● Port 1 ○ Port 2 ○ Port 3
○ Connection through Another Panel

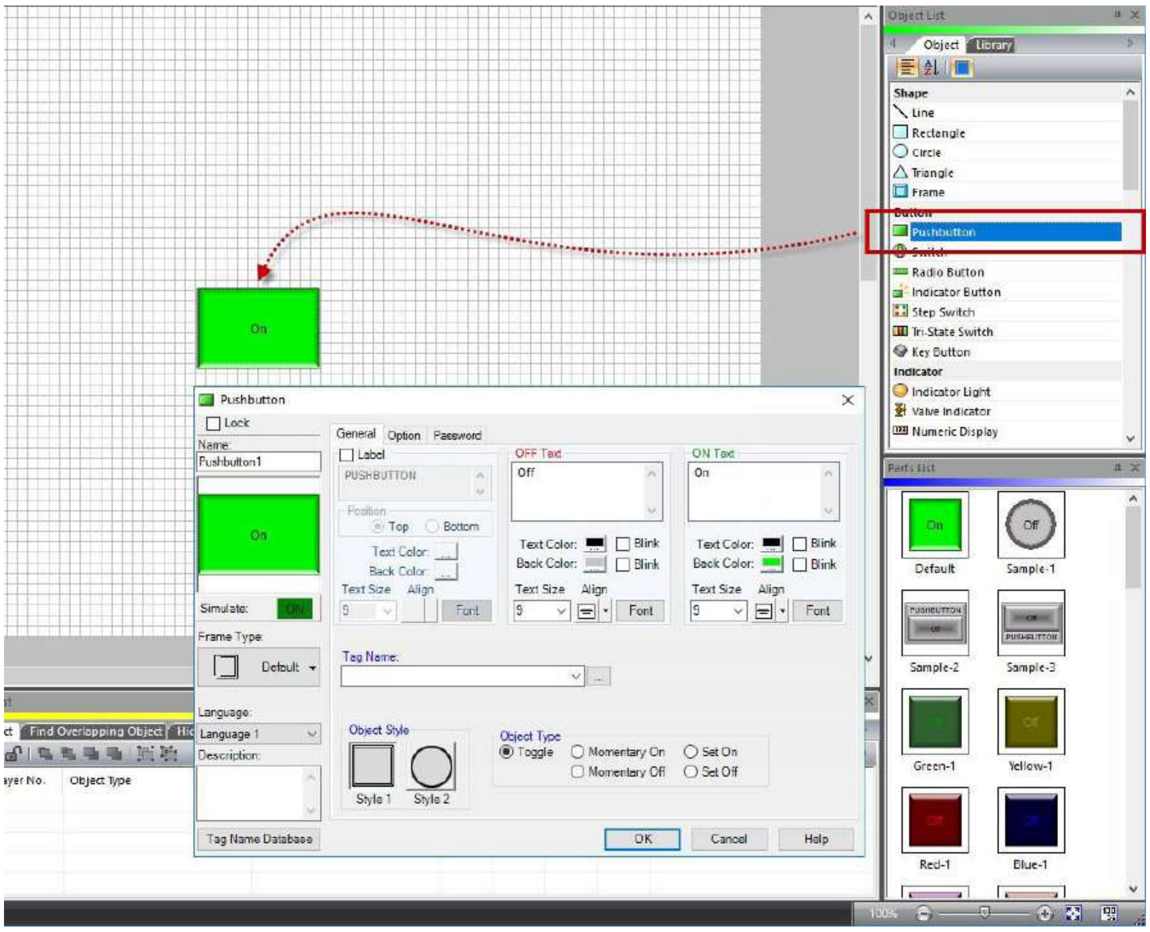
PLC Slave Number(1-247): 2
Baud Rate: 9600
Parity: Odd
Stop Bit: 1
Control RTS: No
Require CTS: No
Select RS485: No
PLC Protocol: RTU
Registers Per Message: 1
Coils Per Message: 1
Timeout Time(1-255) [100ms]: 30
Poll Time(0-30000 msec): 3

Note:
Pin 4 SUPPLIES +5VDC.
Ensure your cable has
no connection on Pin 4.

11. Save the project.

Step 2: Creating Screen Objects

1. Place a pushbutton object on the screen by selecting it from the Object List section and dragging it to the desired position. When this is done, the Pushbutton configuration window will pop up.



Pushbutton

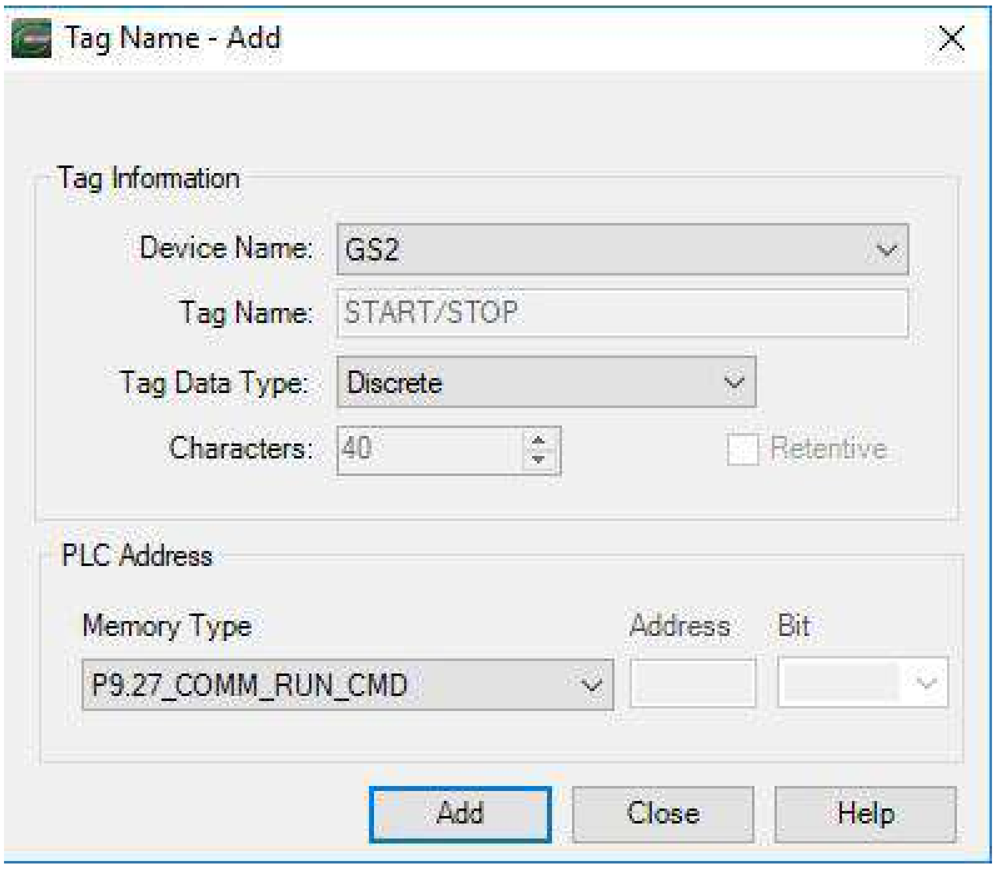
□ Lock
Name: Pushbutton1
Simulate: ON
Frame Type: Default
Language: Language 1
Description:

General tab:
□ Label: PUSHBUTTON
Position: ● Top ○ Bottom
OFF Text: Off — Text Color, Back Color, □ Blink, □ Blink, Text Size: 9
ON Text: On — Text Color, Back Color, □ Blink, □ Blink, Text Size: 9
Tag Name:
Object Style: Style 1, Style 2
Object Type: ● Toggle ○ Momentary On ○ Momentary Off ○ Set On ○ Set Off

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

2. The Pushbutton window is used to customize the object. In this case, we need to associate the pushbutton with the start/stop command on the GS2. Looking in the GS2 User Manual, it says that the start/stop command is parameter P9.27.

Use the Tag Name "START/STOP" for this example. Type "START/STOP" in the Tag Name field and then press "Enter" (on the keyboard). The Tag Name - Add window will then be displayed.



Tag Name - Add

Tag Information

Device Name: GS2

Tag Name: START/STOP

Tag Data Type: Discrete

Characters: 40

□ Retentive

PLC Address

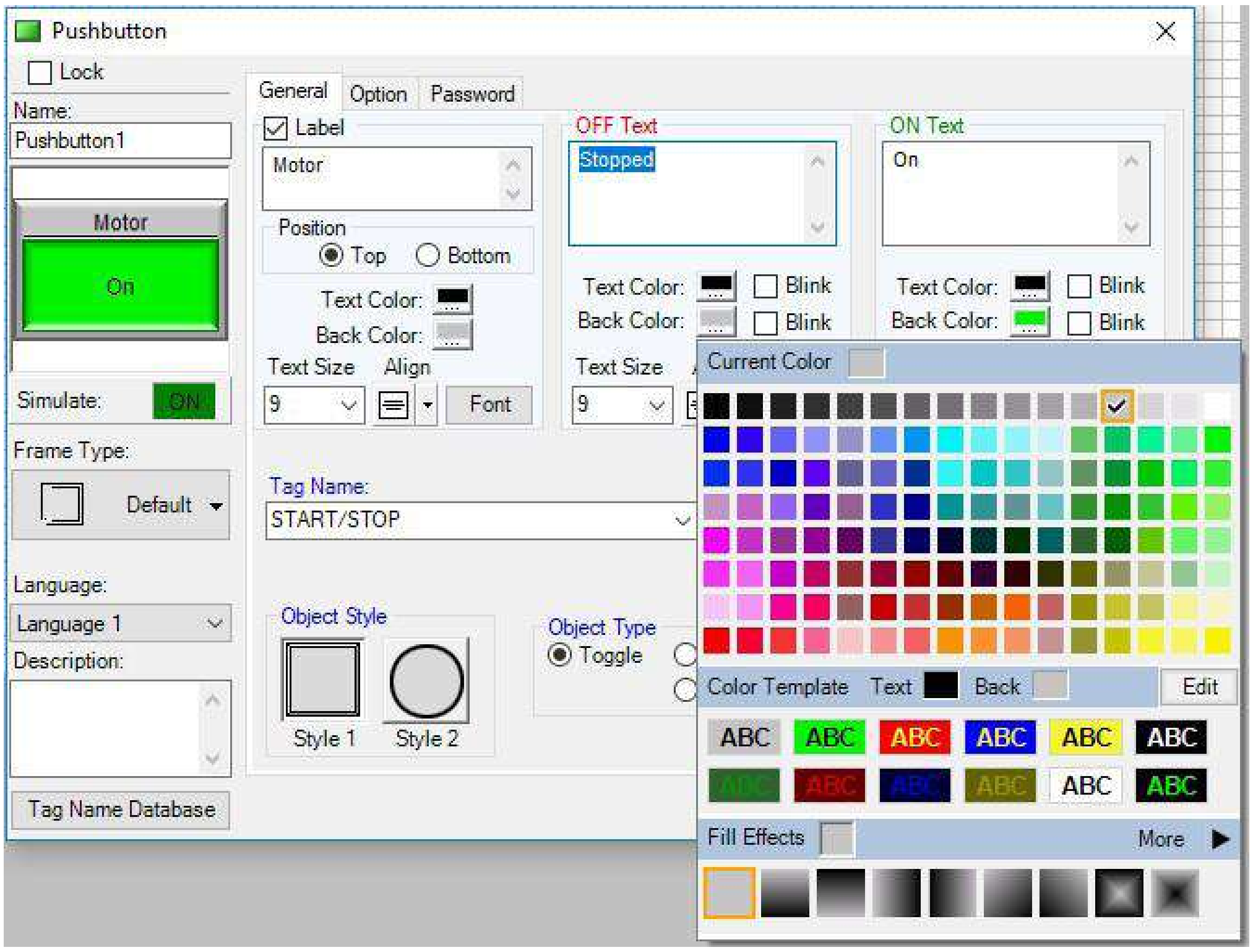
Memory Type: P9.27_COMM_RUN_CMD

Address:

Bit:

- In the Tag Name – Add window, Tag Information section perform the following:
  - Using the down arrow, select Device Name "GS2".
  - Using the down arrow, select Tag Data Type "Discrete" if not already selected.
- In the Tag Name – Add window, PLC Address section perform the following:
  - Using the down arrow, select P9.27 _COMM_RUN_CMD for the Memory Type. This is the start/stop command for the drive as found in the manual previously.
- Click the Add button to save and close the New Tag - Add window and return to the Pushbutton object window.

3. Customize the START/STOP pushbutton as follows:
   - Click the Label check box and replace the "PUSHBUTTON" text with "Motor".
   - Replace the OFF text "Off" with "Stopped".
   - Change the OFF Back Color (in the OFF Text block) to red by clicking on the color button and selecting from the palette.



Pushbutton

□ Lock

Name: Pushbutton1

Simulate: ON

Frame Type: Default

Language: Language 1

Description:

General | Option | Password

☑ Label: Motor

Position: ● Top ○ Bottom

Text Color / Back Color; Text Size: 9; Align

OFF Text: Stopped — Text Color / Back Color, □ Blink, □ Blink; Text Size: 9

ON Text: On — Text Color / Back Color, □ Blink, □ Blink

Tag Name: START/STOP

Object Style: Style 1, Style 2

Object Type: ● Toggle

- Replace the “On” text with “RUN”.
- Click on the OK button to save the Pushbutton object to the project and close the Pushbutton window. There are a variety of options to create a more effective object for your application: you can change the size of the fonts, as well as the fonts themselves, you can make blinking condition, change the style, etc.

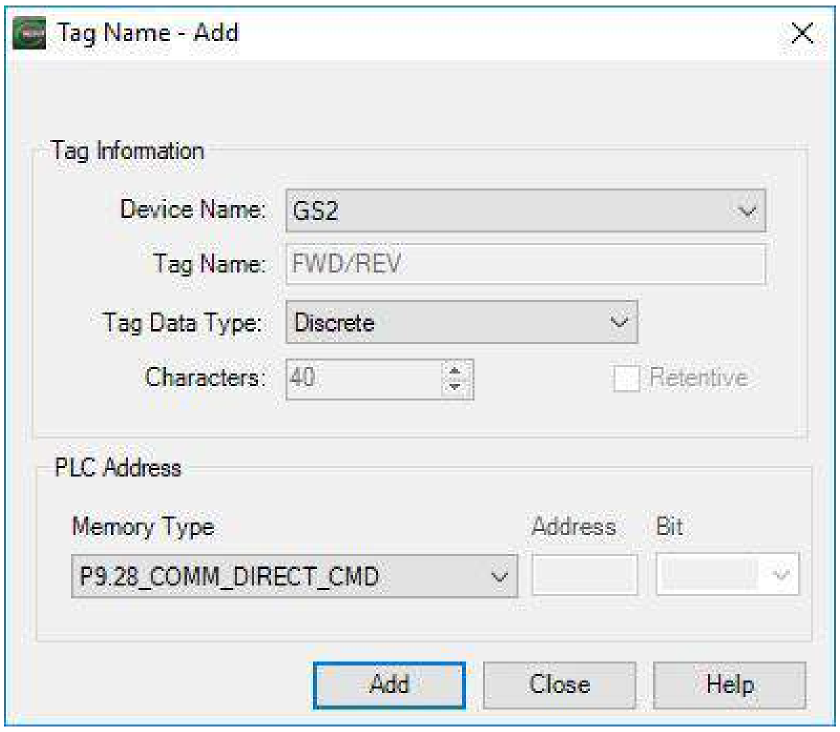
4. Customize the FWD/REV Pushbutton

In order to create the FWD/REV button, you can follow the same procedure as the START/STOP pushbutton above or simply copy and paste the object just created.

A copy will be performed for this example:

- Click once on the Motor Pushbutton object and then copy/paste the object using the CTRL+C and CTRL+V technique or right click for the object menu.
- Move the new Pushbutton object to the desired position. Then, you can change the data by editing the object. Double click on the Pushbutton object to open its configuration window.
- Type over the Tag Name with the new Tag Name “FWD/REV” and then press “Enter” (on the keyboard). The Tag Name - Add window will then be displayed.
- The Device Name and Tag Data Type field should be populated with the correct information. Simply select the drop-down arrow for the Memory Type and choose the drive parameter for direction. Looking in the GS2 User Manual, it says there that the direction parameter is P9.28 so we will select the 9.28_COMM_DIRECT_CMD option. Click Add to close the Tag Name - Add window.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs



Tag Name - Add

Tag Information

Device Name: GS2

Tag Name: FWD/REV

Tag Data Type: Discrete

Characters: 40

□ Retentive

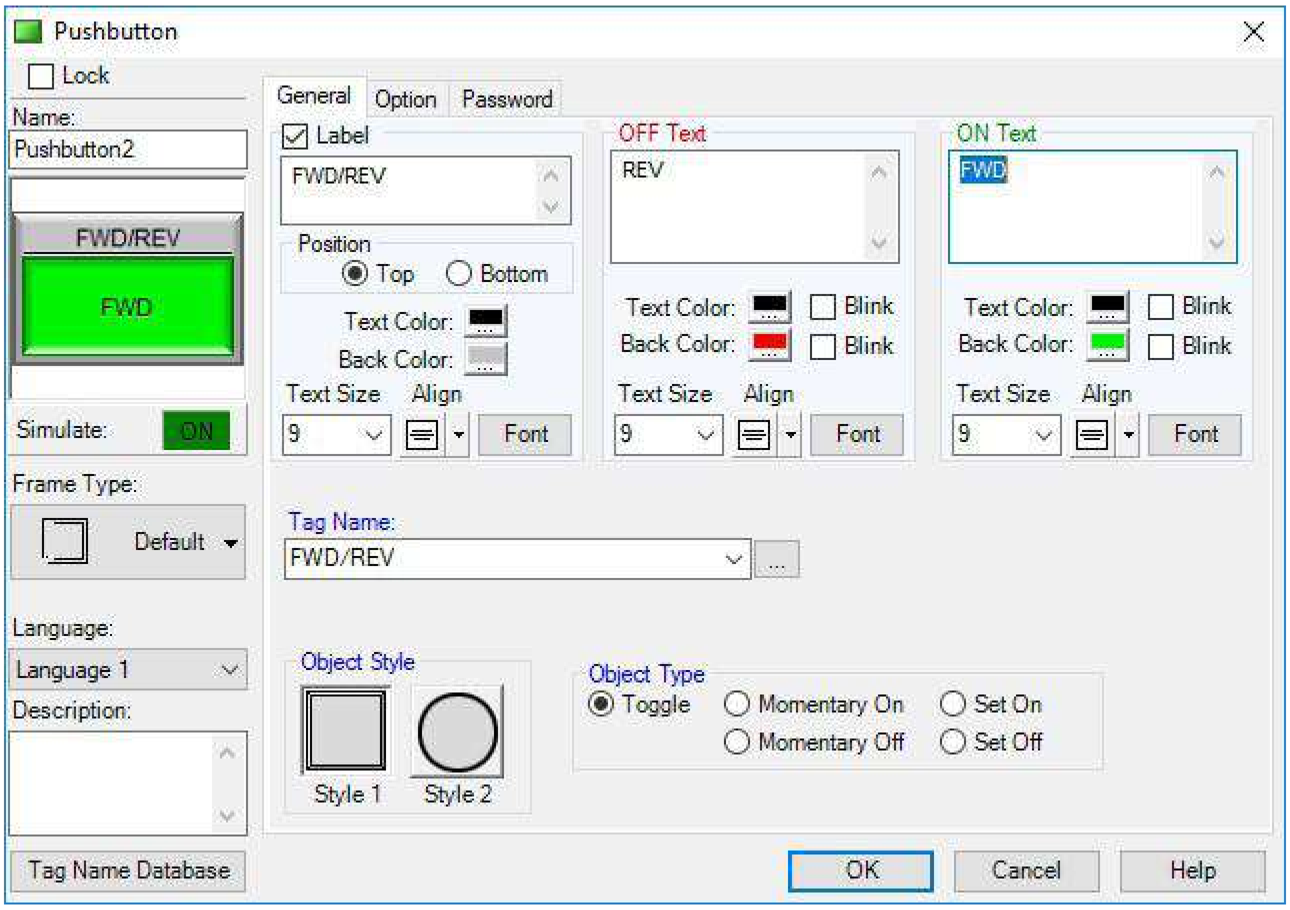
PLC Address

Memory Type: P9.28_COMM_DIRECT_CMD

Address:

Bit:

- Once the pushbutton object window is displayed, for the Label, replace the "Motor" text with "FWD/REV". Replace the OFF text "Stopped" with "REV". Replace the ON text "RUN" with "FWD".



Pushbutton

□ Lock

Name: Pushbutton2

Simulate: ON

Frame Type: Default

Language: Language 1

Description:

☑ Label: FWD/REV

Position: ● Top ○ Bottom

Text Color: ■ Back Color: ■

Text Size: 9 Align

OFF Text: REV

Text Color: ■ □ Blink

Back Color: 🟥 □ Blink

Text Size: 9 Align

ON Text: FWD

Text Color: ■ □ Blink

Back Color: 🟩 □ Blink

Text Size: 9 Align

Tag Name: FWD/REV

Object Style: Style 1 Style 2

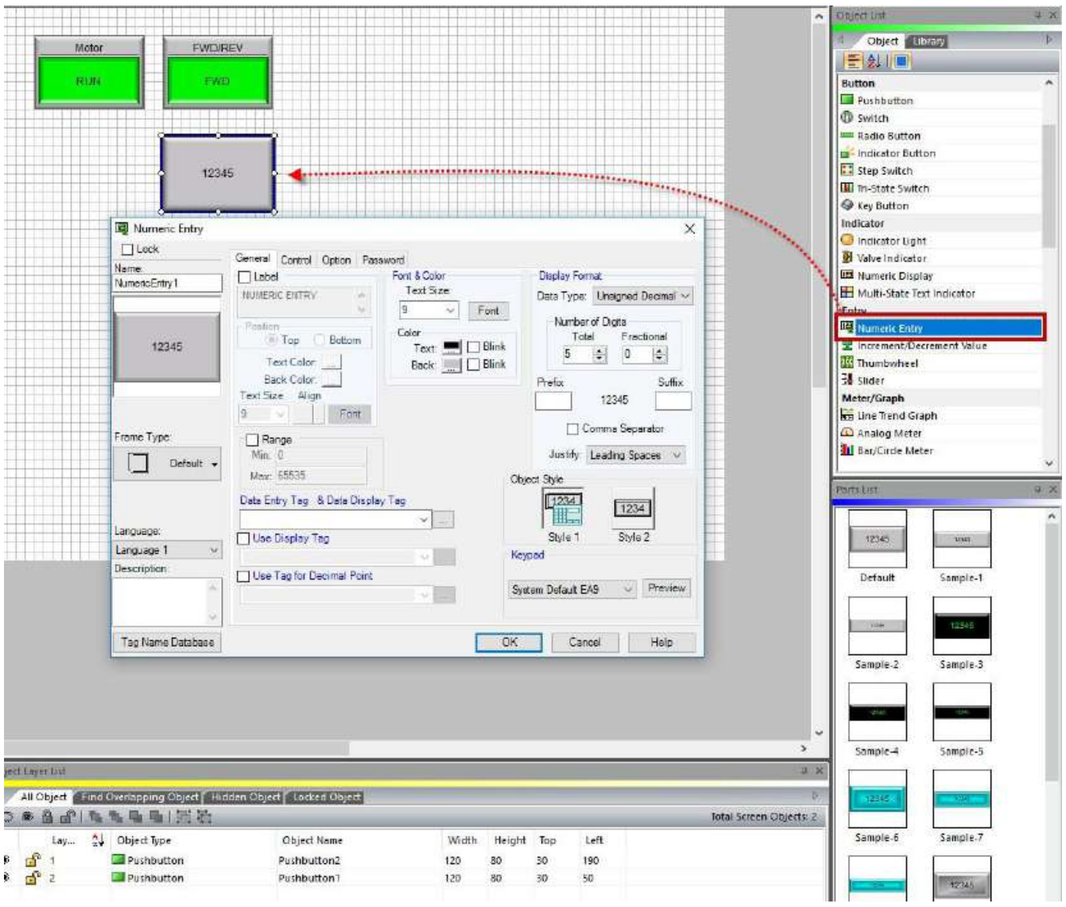
Object Type: ● Toggle ○ Momentary On ○ Set On ○ Momentary Off ○ Set Off

- Click on the OK button to save the pushbutton object to the project and close the Pushbutton window.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

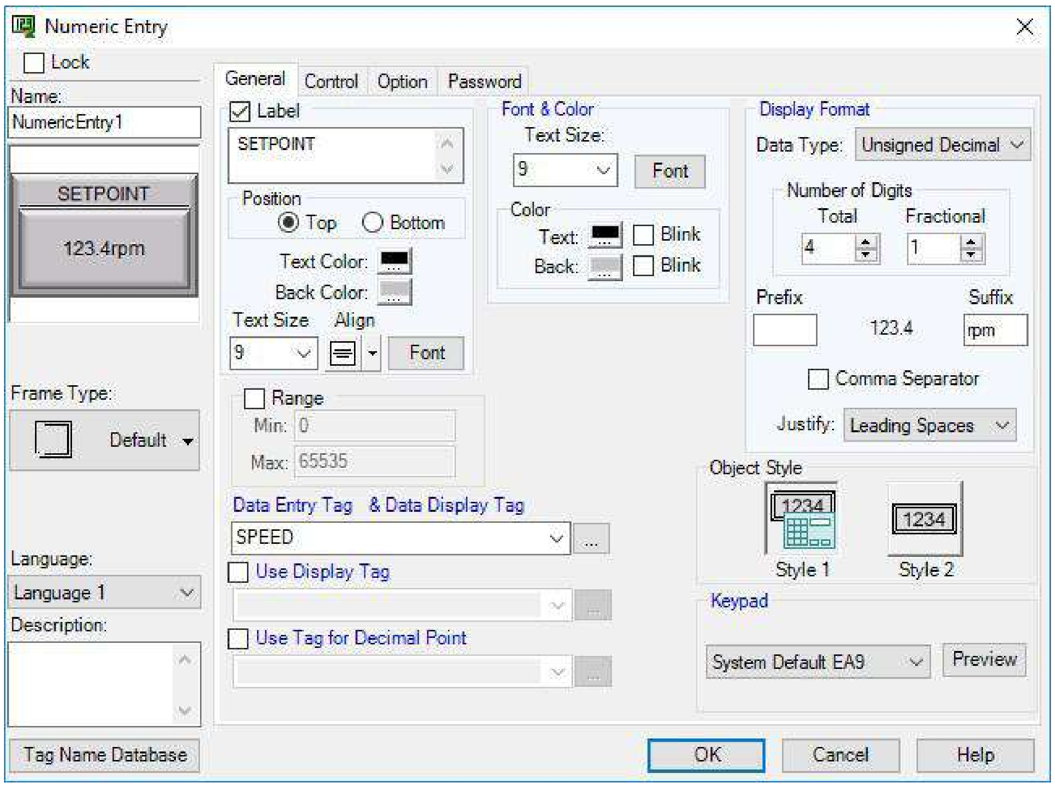
5. Customize the Setpoint Button (Numeric Entry Object)
Now we are ready to create the Numeric Entry object, to set the desired motor speed.

- Place a Numeric Entry object on the screen by selecting it from the Object List section and dragging it to the desired position. When this is done, the Numeric Entry configuration window will pop up.



From the GS2 User Manual, we find that the GS2 accepts frequency as a setpoint, with an unsigned 16-bit decimal format. This is parameter P9.26.

For our tutorial project, we will configure the following fields:
Numeric Entry Window / General Tab:



Numeric Entry

□ Lock

Name: NumericEntry1

SETPOINT 123.4rpm

Frame Type: Default

Language: Language 1

Description:

☑ Label: SETPOINT

Position: ● Top ○ Bottom

Text Color:

Back Color:

Text Size: 9 Align

□ Range Min: 0 Max: 65535

Data Entry Tag & Data Display Tag: SPEED

□ Use Display Tag

□ Use Tag for Decimal Point

Font & Color

Text Size: 9

Color Text: □ Blink

Back: □ Blink

Display Format

Data Type: Unsigned Decimal

Number of Digits Total: 4 Fractional: 1

Prefix: Suffix: rpm

123.4

□ Comma Separator

Justify: Leading Spaces

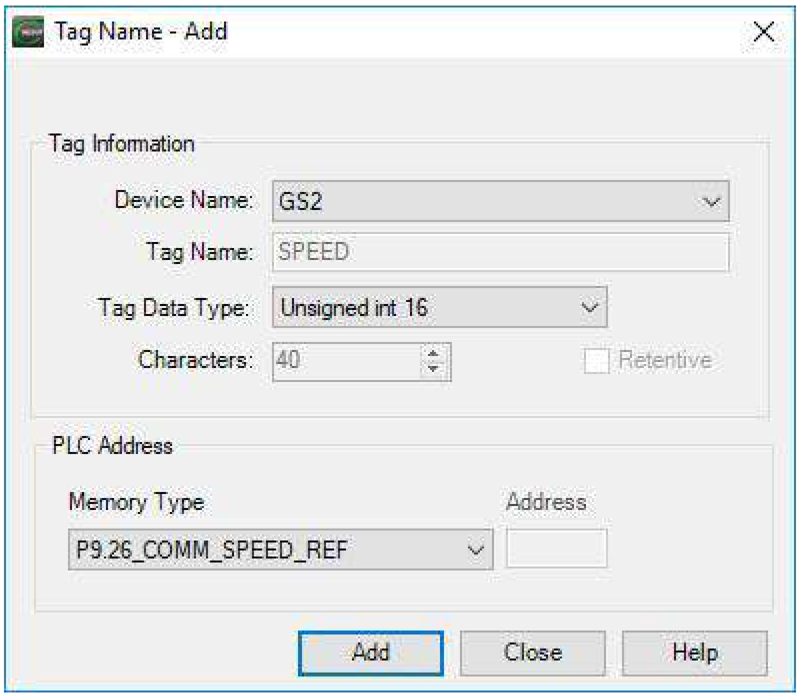
Object Style: Style 1 1234 Style 2 1234

Keypad: System Default EA9

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

- Click the Label check box and replace the NUMERIC ENTRY default label text with "SETPOINT".
- Display Format: Data Type; Using the down arrow, select "Unsigned Decimal" if not already selected.
- Display Format: Number of Digits; Set Total = 4 and Fractional = 1.
- For Suffix enter "rpm".
- For the Data Entry Tag & Data Display Tag: enter "SPEED", then press "Enter". The Tag Name - Add window will then be displayed.

Tag Name - Add Window:



Tag Name - Add

Tag Information

Device Name: GS2

Tag Name: SPEED

Tag Data Type: Unsigned int 16

Characters: 40

□ Retentive

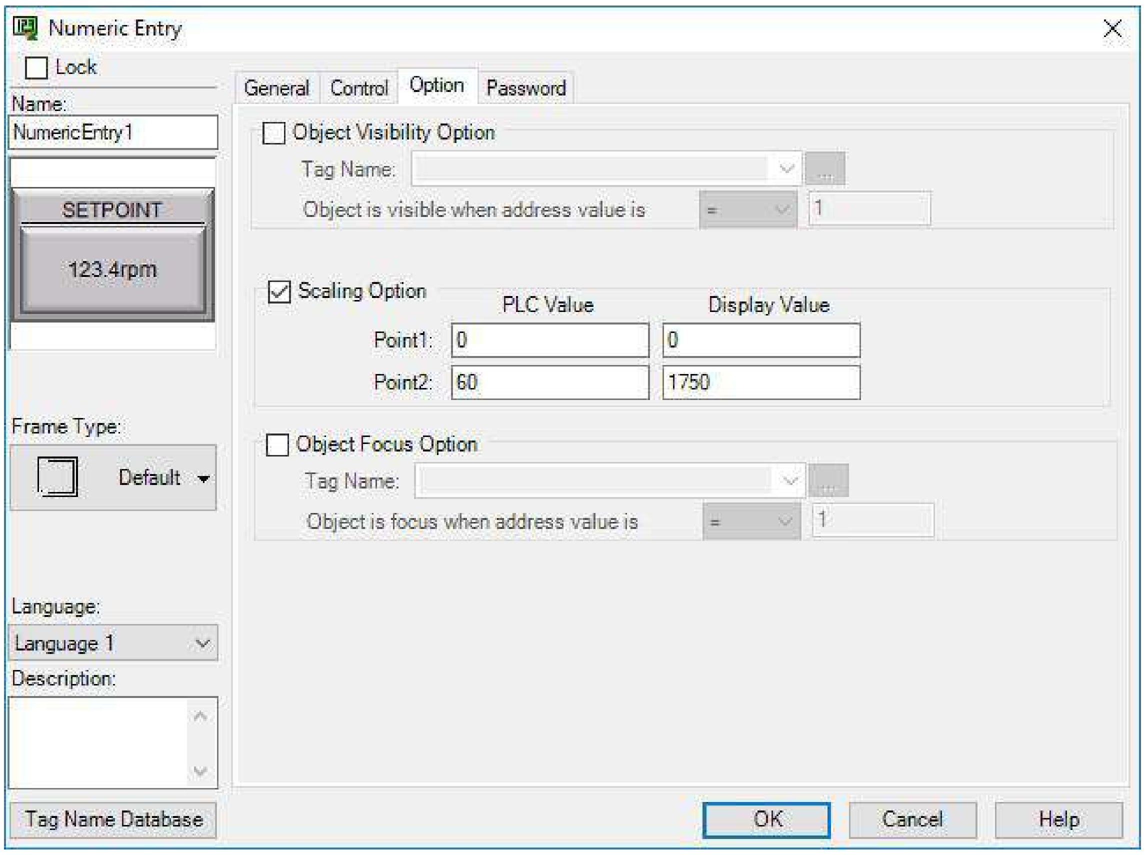
PLC Address

Memory Type: P9.26_COMM_SPEED_REF

Address:

- Tag Information; Device Name: "GS2".
- Tag Information; Tag Data Type: Using the down arrow, select "Unsigned int 16".
- PLC Address; Memory Type: Using the down arrow, select "P9.26_COMM_SPEED_REF".
- Click the Add button to save and close the New Tag - Add window and return to the Numeric Entry object window.

Numeric Entry Window / Option tab:



Numeric Entry

□ Lock

Name: NumericEntry1

SETPOINT
123.4rpm

Frame Type: Default

Language: Language 1

Description:

General | Control | Option | Password

□ Object Visibility Option

Tag Name:

Object is visible when address value is = 1

☑ Scaling Option

	PLC Value	Display Value
Point1:	0	0
Point2:	60	1750

□ Object Focus Option

Tag Name:

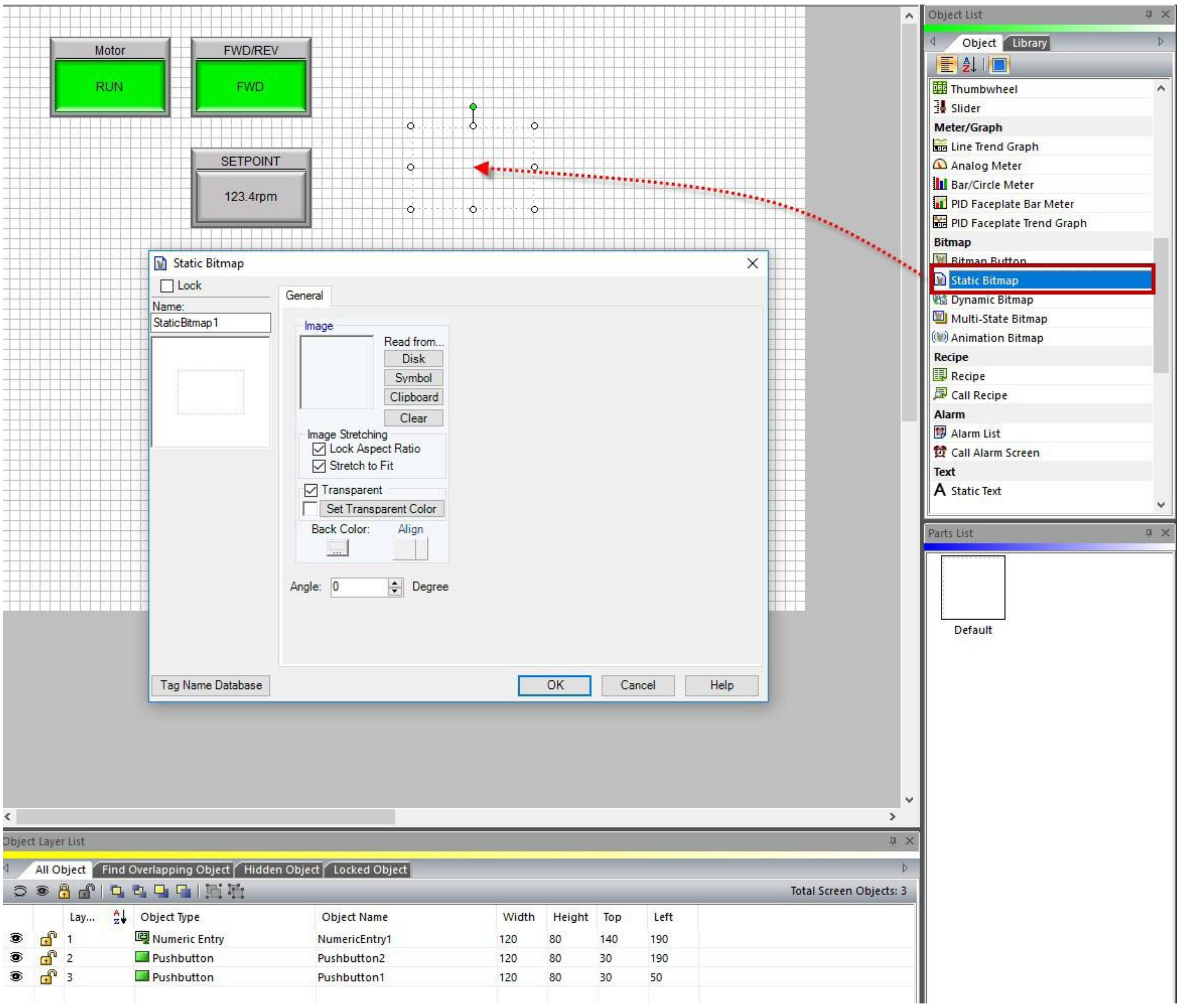
Object is focus when address value is = 1

- Select Scaling Option check box.
- For Point1: PLC Value "0" (Hz) will correspond to Display Value "0" (rpm).
- For Point2: PLC Value "60" (Hz) will correspond to Display Value "1750" (rpm.)
- Click on the OK button to save the Numeric Entry object to the project and close the Numeric Entry window.

6. Create a Bitmap Object

Now we are ready to create the Bitmap object, which can be used to place a picture of your machine, or a mimic diagram of your process, or even a logo.

- Place a Static Bitmap object on the screen by selecting it from the Object List section and dragging it to the desired position. When this is done, the Static Bitmap configuration window will pop up.



Static Bitmap

☐ Lock

Name: StaticBitmap1

Image Stretching
☑ Lock Aspect Ratio
☑ Stretch to Fit

☑ Transparent
☐ Set Transparent Color

Angle: 0 Degree

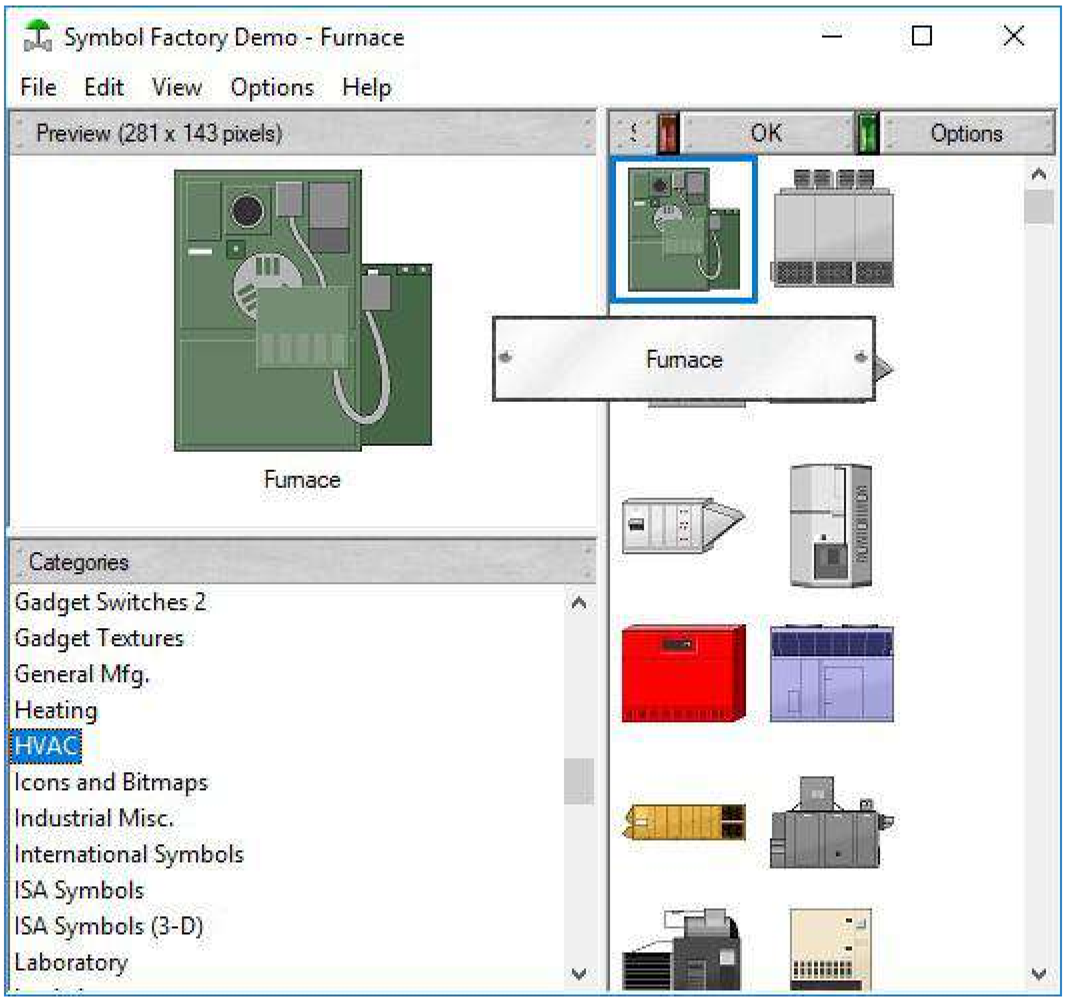
Object Layer List

Total Screen Objects: 3

Lay...	Object Type	Object Name	Width	Height	Top	Left
1	Numeric Entry	NumericEntry1	120	80	140	190
2	Pushbutton	Pushbutton2	120	80	30	190
3	Pushbutton	Pushbutton1	120	80	30	50

- On the Image box, select the Symbol button to open the available images. For this tutorial we will use an HVAC, Furnace Bitmap image.
- From the Categories list, select "HVAC" and click on the first image (top left). The Static Bitmap window will now display the selected image.

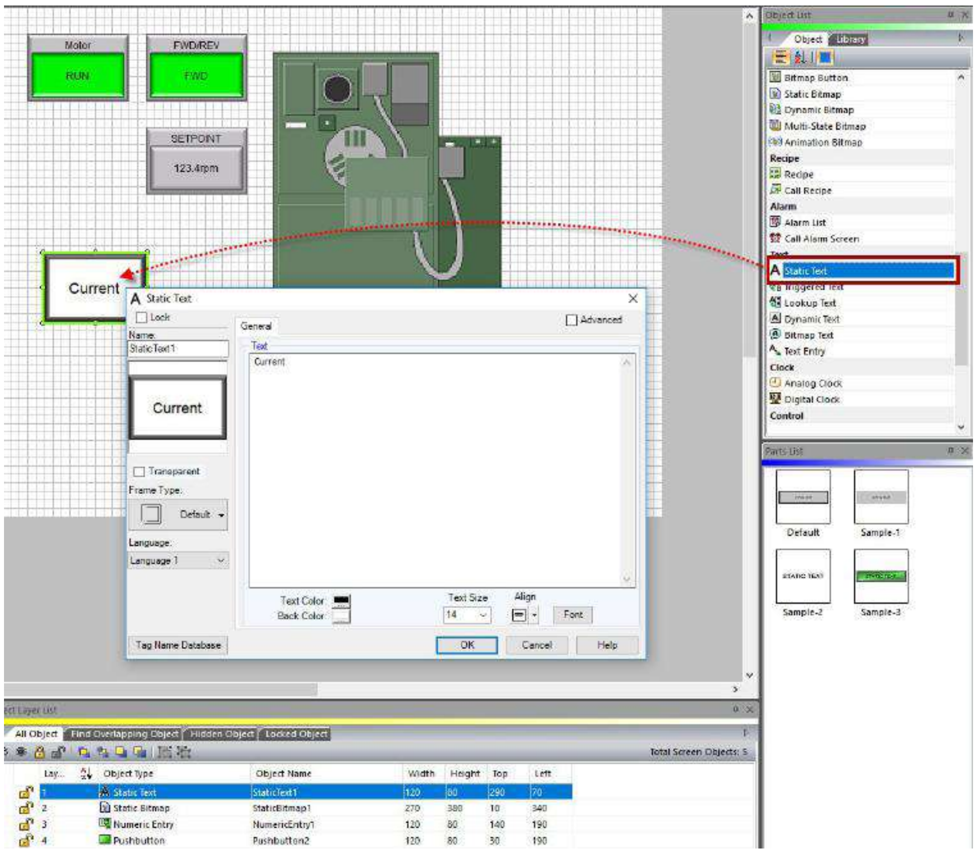
Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs



- Close the Symbol Factory Demo window.
- Click on the OK button to save the Static Bitmap object to the project and close the Static Bitmap window.
- Resize the bitmap on the screen as you see fit.

7. Create the Current Display tag (Static Text Object)

- Place a Static Text object on the screen by selecting it from the Object List section and dragging it to the desired position. When this is done, the Static Text configuration window will pop up.

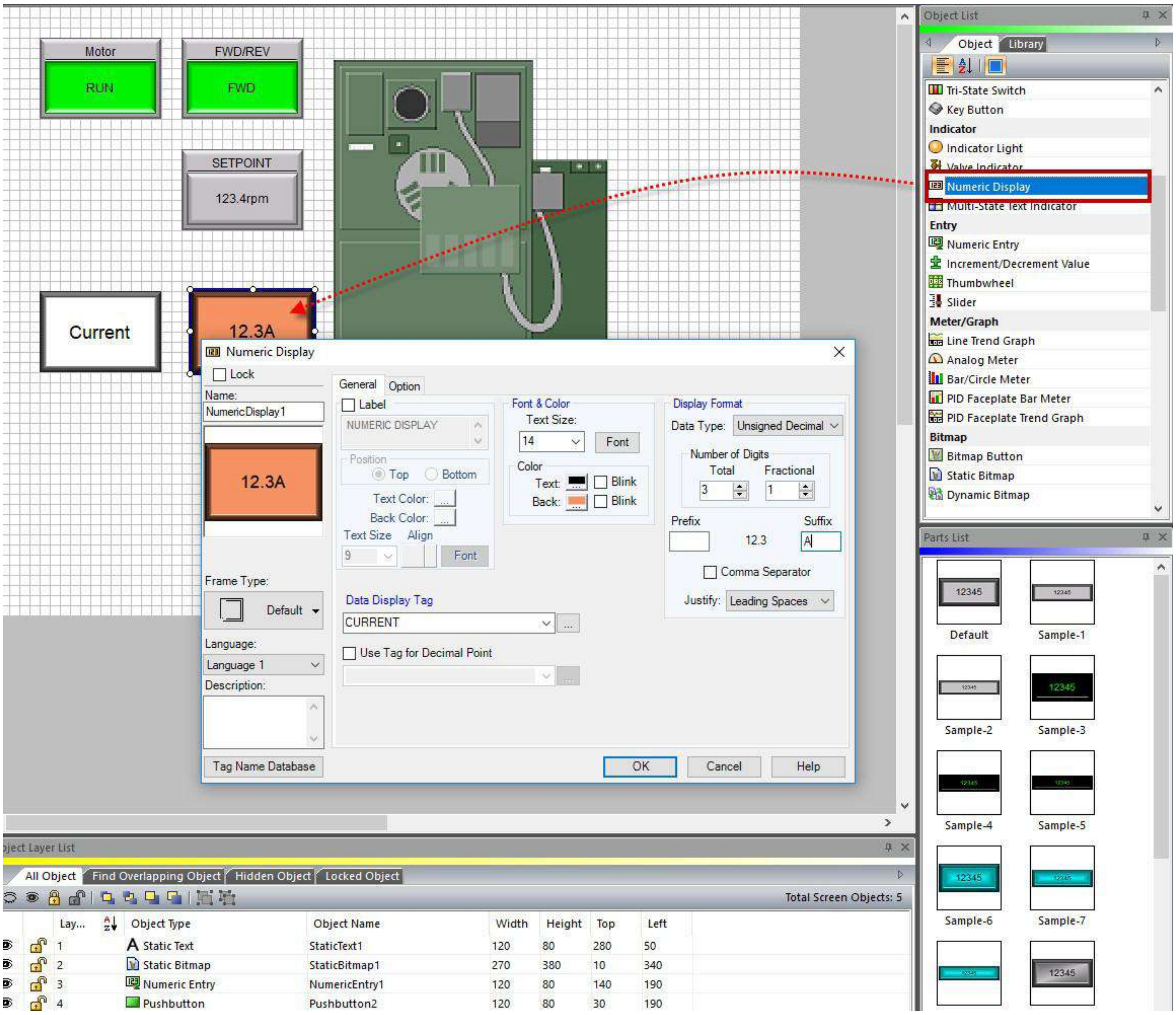


Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

- In the Text box, replace "STATIC TEXT" with "Current".
- Open the Back Color palette and select white for the background.
- Select Text Size: of "14".
- Click on the OK button to save the Static Text object to the project and close the Static Text window.

8. Create the Current Display Window (Numeric Display Object)

- Place a Numeric Display object on the screen by selecting it from the Object List section and dragging it to the desired position. When this is done, the Numeric Display configuration window will pop up.



Numeric Display

☐ Lock

Name: NumericDisplay1

General | Option

☐ Label

NUMERIC DISPLAY

Position: ● Top ○ Bottom

Text Color: ...

Back Color: ...

Text Size: 9 Align Font

Font & Color

Text Size: 14 Font

Color

Text: ■ ☐ Blink

Back: ■ ☐ Blink

Display Format

Data Type: Unsigned Decimal

Number of Digits

Total: 3 Fractional: 1

Prefix: 12.3 Suffix: A

☐ Comma Separator

Justify: Leading Spaces

Data Display Tag: CURRENT

☐ Use Tag for Decimal Point

Frame Type: Default

Language: Language 1

Description:

Tag Name Database | OK | Cancel | Help

Object Layer List

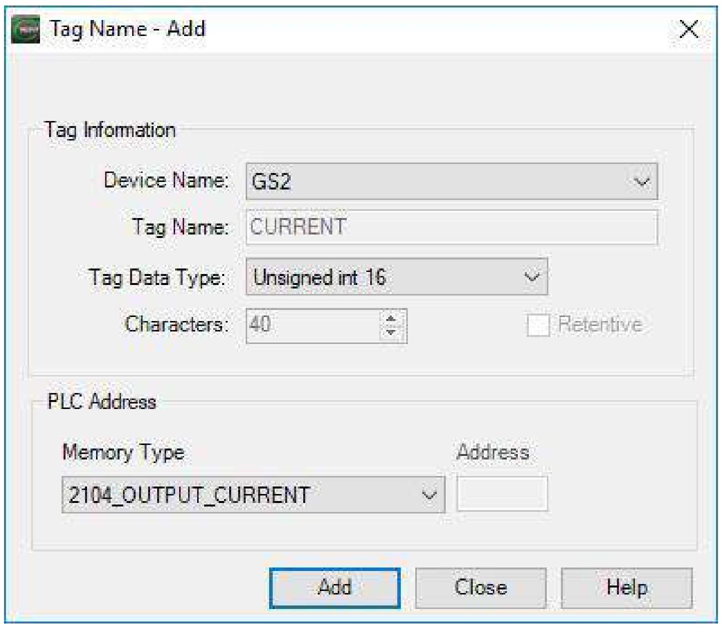
All Object | Find Overlapping Object | Hidden Object | Locked Object

Total Screen Objects: 5

Lay...	Object Type	Object Name	Width	Height	Top	Left
1	Static Text	StaticText1	120	80	280	50
2	Static Bitmap	StaticBitmap1	270	380	10	340
3	Numeric Entry	NumericEntry1	120	80	140	190
4	Pushbutton	Pushbutton2	120	80	30	190

- Set the Font and Color; Text Size to 14.
- Open the Font and Color; Back palette and select "light orange".
- Using the down arrow, select Display Format; Data Type to Unsigned Decimal.
- Set Display Format; Number of Digits: Total to 3 and Fractional to 1.
- Set Display Format; Suffix to "A".
- For the Data Display Tag type "CURRENT", then press "Enter". The Tag Name - Add window will then be displayed.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs



Tag Name - Add

Tag Information

Device Name: GS2

Tag Name: CURRENT

Tag Data Type: Unsigned int 16

Characters: 40 □ Retentive

PLC Address

Memory Type: 2104_OUTPUT_CURRENT

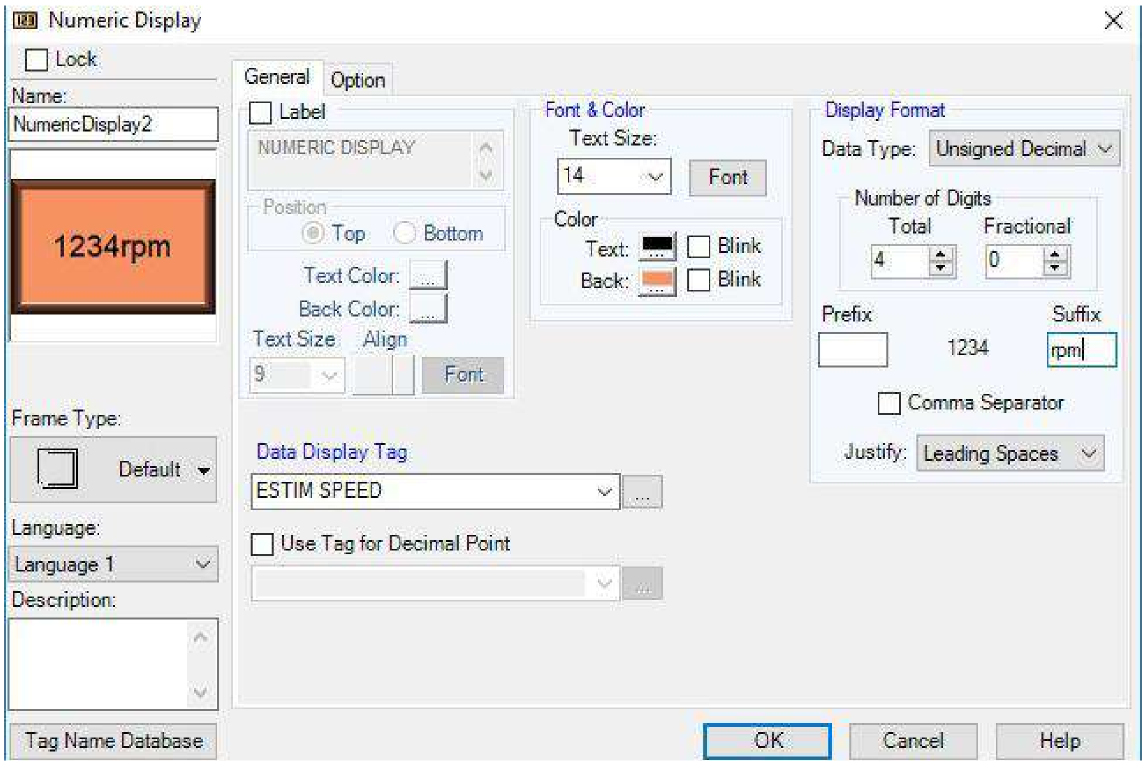
Address:

- From the GS2 User Manual we find the Output Current tag data type is Unsigned int 16, at address 2104. Configure the Tag Name - Add window with the following: Device Name = "GS2", Tag Data Type = "Unsigned int 16", Memory Type = "2104_OUTPUT_CURRENT".
- Click on the Add button then click on the OK button to save the Numeric Display object to the project and close the Numeric Display window.

9. Creating the Estimated Speed Tag and Display window (Static Text and Numeric Display Objects)

In order to create the Estimated Speed display objects we will copy the Static Text and Numeric Display objects created previously.

- Select both objects, then copy/paste duplicates on the screen and move the two new objects to a suitable location.
- Double click the new Static Text object and replace the "Current" text with "Estimated Speed".
- Click OK to save the changes and close the Static Text window.
- Double click on the new Numeric Entry object to open the Numeric Display window.



Numeric Display

□ Lock

Name: NumericDisplay2

1234rpm

Frame Type: Default

Language: Language 1

Description:

General | Option

□ Label: NUMERIC DISPLAY

Position: ○ Top ○ Bottom

Text Color:

Back Color:

Text Size: 9 Align:

Font & Color

Text Size: 14

Color Text: □ Blink

Back: □ Blink

Display Format

Data Type: Unsigned Decimal

Number of Digits: Total 4 Fractional 0

Prefix: Suffix: rpm

1234

□ Comma Separator

Justify: Leading Spaces

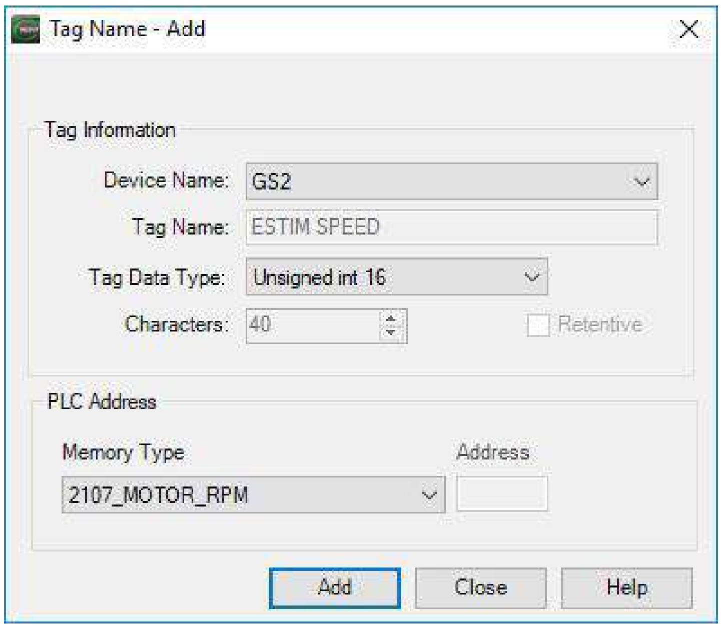
Data Display Tag: ESTIM SPEED

□ Use Tag for Decimal Point

Set the Font and Color; Text Size to 14.

- Set Display Format; Number of Digits: Total to 4 and Fractional to 0.
- Set Display Format; Suffix to "rpm"
- Replace the Data Display Tag with "ESTIM SPEED" then press "Enter". The Tag Name - Add window will then be displayed.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs



Tag Name - Add

Tag Information

Device Name: GS2

Tag Name: ESTIM SPEED

Tag Data Type: Unsigned int 16

Characters: 40 □ Retentive

PLC Address

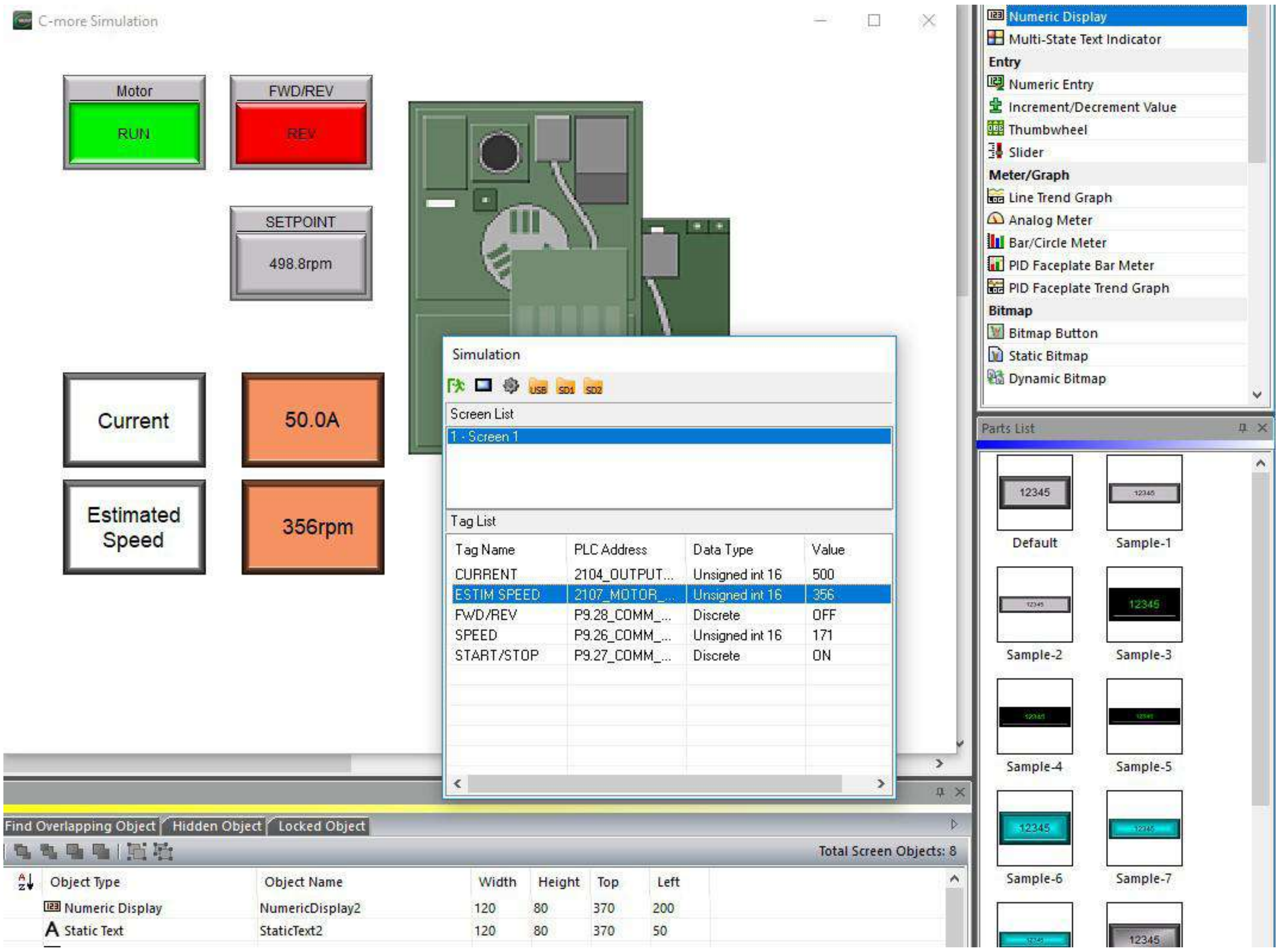
Memory Type: 2107_MOTOR_RPM

Address:

- From the GS2 User Manual, we find the Motor RPM tag data type is Unsigned int 16, at address 2107. Configure the Tag Name – Add window with the following: Device Name = "GS2", Tag Data Type = "Unsigned int 16", Memory Type = "2107_MOTOR_RPM".
- Click on the Add button then click on the OK button to save the Numeric Display object to the project and close the Numeric Display window.

10. Save Project and Send to Panel

All that is left to do now is to save the project and transfer it to the panel. With C-more you also have the option to simulate the project to test your objects before the transfer. See the Simulation chapter for more information.



Tag List

Tag Name	PLC Address	Data Type	Value
CURRENT	2104_OUTPUT...	Unsigned int 16	500
ESTIM SPEED	2107_MOTOR_...	Unsigned int 16	356
FWD/REV	P9.28_COMM_...	Discrete	OFF
SPEED	P9.26_COMM_...	Unsigned int 16	171
START/STOP	P9.27_COMM_...	Discrete	ON

Total Screen Objects: 8

Object Type	Object Name	Width	Height	Top	Left
Numeric Display	NumericDisplay2	120	80	370	200
Static Text	StaticText2	120	80	370	50

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Once the project is transferred, the panel will provide an interface to the GS2 drive and provide simple control of a motor.

For a step-by-step demonstration on how to connect to your C-more HMI and transfer a project check out this video:

Learn More...Watch The Video!

C-more HMI: Ethernet Quick Start

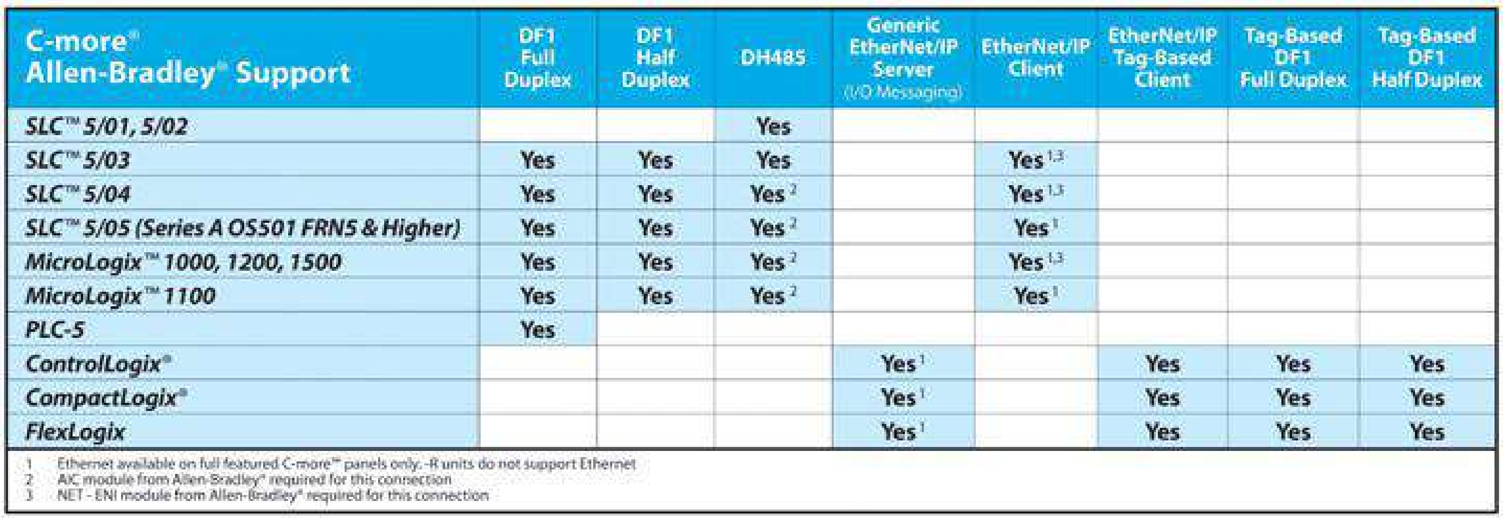
http://n2adc.com/-8bet

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Using C-more with Allen-Bradley PLCs

With the growing popularity of the C-more Operator Interface line, we have had many requests to add PLC protocol drivers. These drivers allow customers with different PLC brands to standardize on the C-more HMI product.

This article takes a look at C-more's PLC drivers and demonstrates how to import/export tags and optimize communication. Due to the abundance of requests, the latest group of drivers added supports the Allen-Bradley family of PLCs, so those are the drivers we will focus on. The new Ethernet drivers allow simple connectivity of multiple C-more panels and/or multiple Allen-Bradley PLCs.



C-more® Allen-Bradley® Support	DF1 Full Duplex	DF1 Half Duplex	DH485	Generic EtherNet/IP Server (I/O Messaging)	EtherNet/IP Client	EtherNet/IP Tag-Based Client	Tag-Based DF1 Full Duplex	Tag-Based DF1 Half Duplex
SLC™ 5/01, 5/02			Yes					
SLC™ 5/03	Yes	Yes	Yes		Yes[1,3]			
SLC™ 5/04	Yes	Yes	Yes[2]		Yes[1,3]			
SLC™ 5/05 (Series A OS501 FRN5 & Higher)	Yes	Yes	Yes[2]		Yes[1]			
MicroLogix™ 1000, 1200, 1500	Yes	Yes	Yes[2]		Yes[1,3]			
MicroLogix™ 1100	Yes	Yes	Yes[2]		Yes[1]			
PLC-5	Yes							
ControlLogix®				Yes[1]		Yes	Yes	Yes
CompactLogix®				Yes[1]		Yes	Yes	Yes
FlexLogix				Yes[1]		Yes	Yes	Yes

[1] Ethernet available on full featured C-more™ panels only. -R units do not support Ethernet
[2] AIC module from Allen-Bradley® required for this connection
[3] NET - ENI module from Allen-Bradley® required for this connection

We also added ControlLogix EtherNet/IP Tag Messaging support. This new feature helps increase productivity by reducing the time often required to map your PLC tag database into another device. This time saving feature allows you to import the RSLogix 5000 L5K file directly, or with just a few clicks of the mouse you can directly enter your ControlLogix or CompactLogix tags from the PLC into a C-more database. No mapping or translation is required.

Since these protocols are widely used with C-more HMI, we have developed some tips on when and how to use these new drivers with C-more panels. To explain this, we will use common Allen-Bradley terminology. More details for the following can also be found on the AutomationDirect Technical Support web site. Look for document number AN-EA-004 on the Technical Notes page.

Compatible Allen-Bradley PLCs

C-more supports five Allen-Bradley PLC drivers: DF1 Full Duplex, DF1 Half Duplex, DH485, EtherNet/IP Client driver and EtherNet/IP Server driver (generic).

The compatibility matrix above illustrates the many choices available to connect Allen-Bradley PLCs to the C-more HMI.

Which Protocol Should I Use?

These descriptions of the five available protocols can assist you in choosing the best one for your application.

DF1 Full Duplex: This driver is used for point-to-point serial connections. If you are connecting only one C-more panel and one Allen-Bradley PLC, this is the most economical method.

DF1 Half Duplex: This is an economical method for connecting one C-more panel to multiple PLCs. Note that since Allen-Bradley PLCs do not use RS-422 or RS-485 for serial connection, an RS-232 to RS-422/485 converter (such as the FA-ISOCON) is required for each PLC.

Generic EtherNet/IP Server: This driver is atypical in that the messaging is initiated by the PLC instead of C-more. The C-more panel is treated basically like a Remote I/O drop. It contains a block of Input Words and Output Words and the PLC either reads or writes this data back and forth. The PLC tags are not directly referenced, as in the EtherNet/IP Client driver. This rare method is sometimes preferred by customers for unique applications.

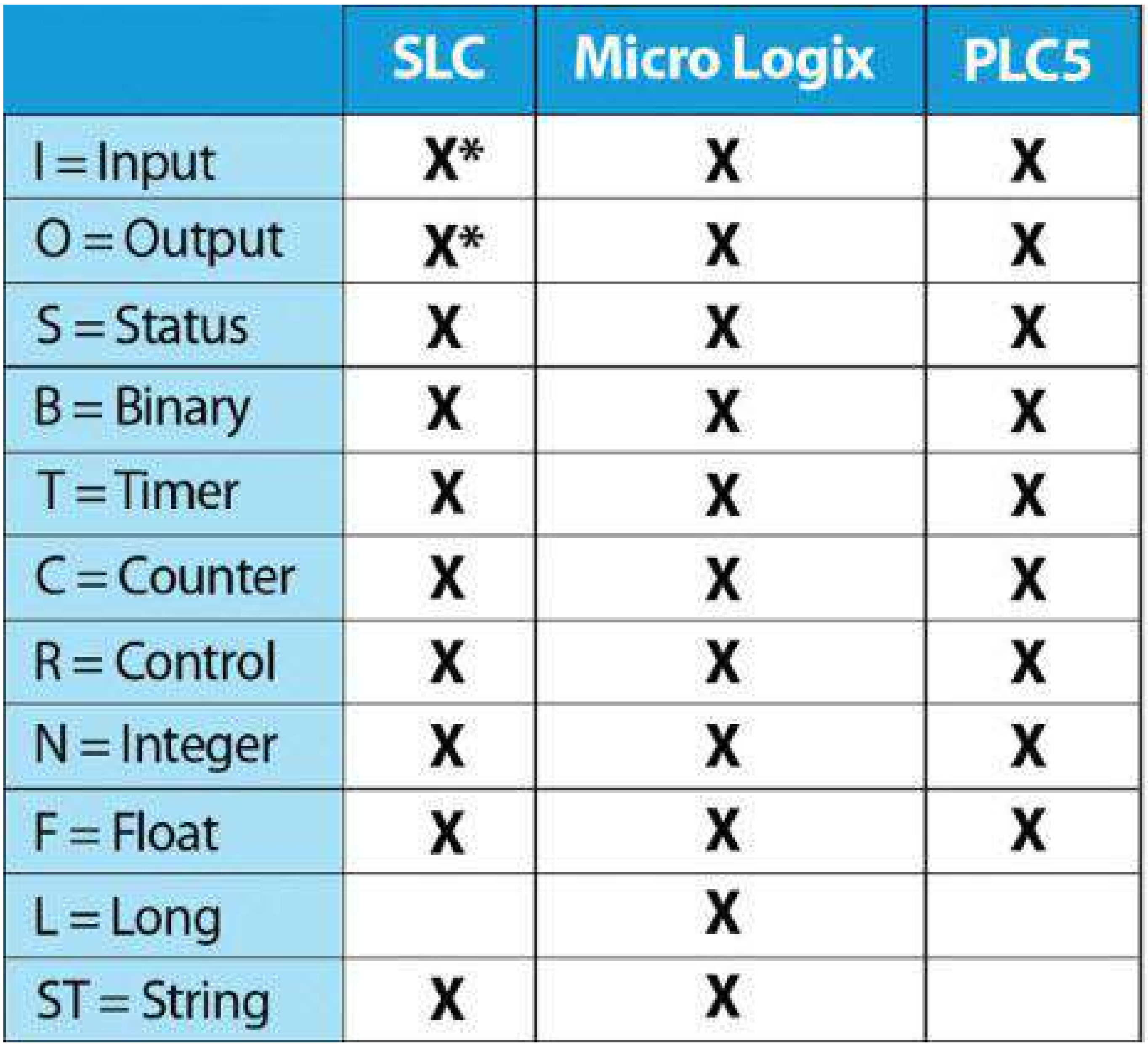
EtherNet/IP Client: This driver is the more typical method of reading and writing data back and forth from the PLC. The C-more panel directly references the PLC tags and initiates the messages going back and forth. If your application involves using multiple C-more panels and/or multiple PLCs, this is the best method for accomplishing this.

DH485: This driver is mostly for legacy systems where it is necessary to add a C-more panel to an existing network. It is possible with DH485 to have multiple C-more panels and multiple PLCs on the same network but the performance levels are much lower than EtherNet/IP. DH485 is not as robust as DF1 or EtherNet/IP.

Supported Data Types

The earlier Allen-Bradley SLC, MicroLogix, and PLC5 support what can be referred to as legacy memory mapping. This mapping is based on fixed file types, whereas the newer Control/Compact/FlexLogix PLCs use what is referred to as a variable tag based memory map. The **C-more** HMI supports both types of mapping.

Compatible Data Type Chart for SLC, MicroLogix, and PLC5



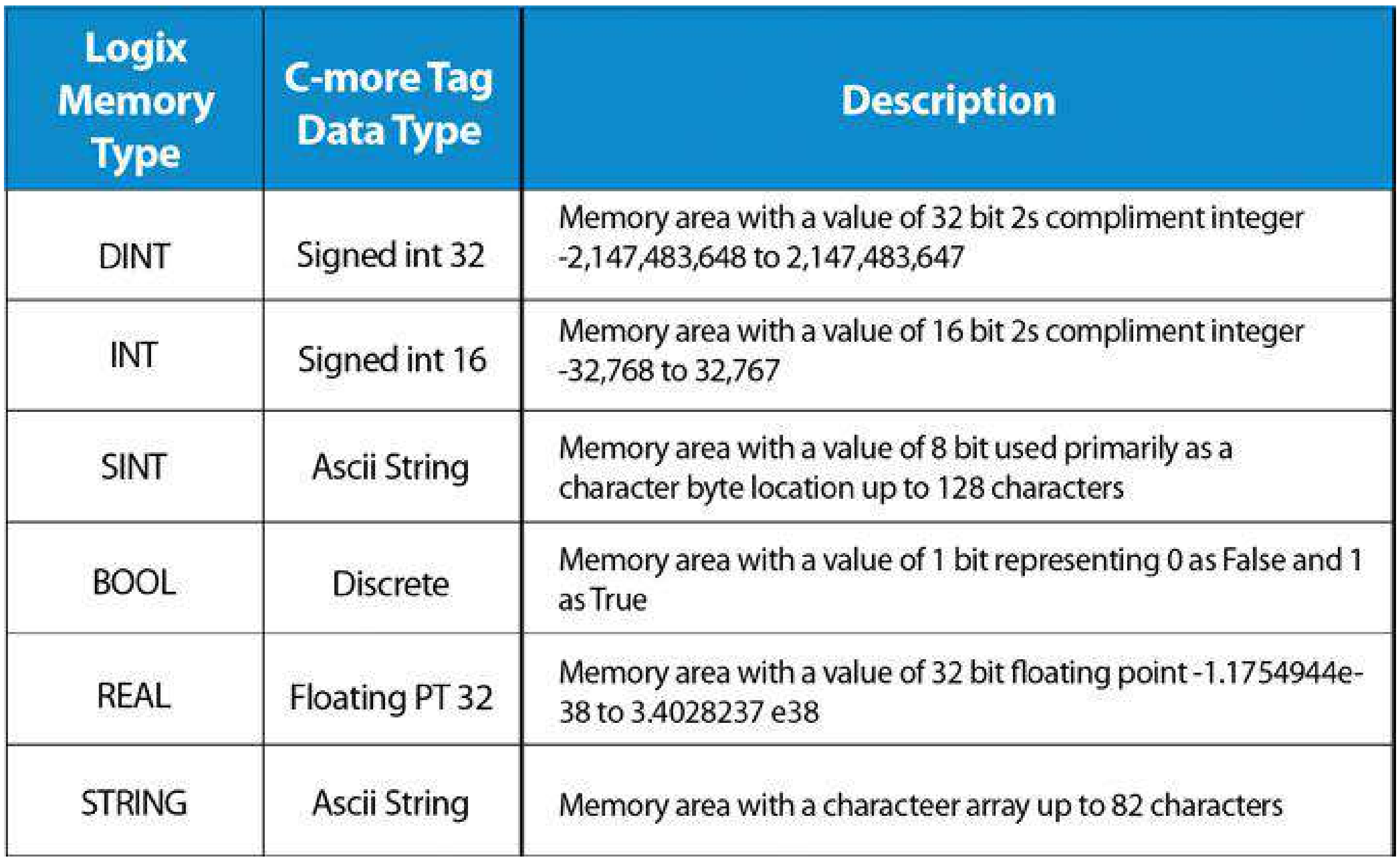
	SLC	Micro Logix	PLC5
I = Input	X*	X	X
O = Output	X*	X	X
S = Status	X	X	X
B = Binary	X	X	X
T = Timer	X	X	X
C = Counter	X	X	X
R = Control	X	X	X
N = Integer	X	X	X
F = Float	X	X	X
L = Long		X	
ST = String	X	X	

** I/O access not supported for SLC5/01 and 5/02*

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Control/Compact/FlexLogix Tag Support

There are different terms used to describe the different data types in the Allen Bradley Logix PLCs. The table below describes how the Allen-Bradley data types are used in the C-more programming software.



Logix Memory Type	C-more Tag Data Type	Description
DINT	Signed int 32	Memory area with a value of 32 bit 2s compliment integer -2,147,483,648 to 2,147,483,647
INT	Signed int 16	Memory area with a value of 16 bit 2s compliment integer -32,768 to 32,767
SINT	Ascii String	Memory area with a value of 8 bit used primarily as a character byte location up to 128 characters
BOOL	Discrete	Memory area with a value of 1 bit representing 0 as False and 1 as True
REAL	Floating PT 32	Memory area with a value of 32 bit floating point -1.1754944e-38 to 3.4028237 e38
STRING	Ascii String	Memory area with a characteer array up to 82 characters

Tag Types

The C-more HMI supports all the tag and memory types used in the Logix PLCs. Each has different characteristics that will determine how it is used in your HMI project.

Atomic Data: Atomic data types are the base data type unit from which all other complex data types are built. The Atomic data types are: BOOL, SINT, INT, DINT and REAL. The syntax for the Atomic data types is simply the tag name itself (no precursors or delimiters). An example would be: Tank1_Level.

All Atomic data types are addressable in the C-more programming software and are importable via the L5K file.

Pre-Defined Data: Pre-defined data types are complex data types that are made up of one or more Atomic data types, user-defined data types or arrays. They are commonly called “structures” in other programming languages such as C. The pre-defined data types already exist in RS Logix 5000 with every new project that is created. One of the most common pre-defined data types is the TIMER data type.

A TIMER data type comprises these members:

TIMER:
PRE (preset member that is a DINT atomic data type)
.ACC (accumulated member that is a DINT atomic data type)
.EN (enabled member that is a BOOL atomic data type)
.TT (timer timing member that is a BOOL atomic data type)
.DN (done member that is a BOOL atomic data type)
.FS (first scan member that is a BOOL atomic data type)
.LS (last scan member that is a BOOL atomic data type)
.OV (over member that is a BOOL atomic data type)
.ER (error member that is a BOOL atomic data type)

So if you create a TIMER called Cycle in the PLC, you will have these tags:
Cycle.PRE, Cycle.ACC, Cycle.EN, Cycle.TT, Cycle.DN, Cycle.FS, Cycle.LS, Cycle.OV and Cycle.ER.

There are hundreds of pre-defined data types available in the RS Logix 5000 software.

All pre-defined data types are addressable in the C-more programming software. The pre-defined data types can be imported via the L5K file, however the configuration file "L5KPreDefine.txt" must be modified for this purpose. For more details, refer to application document AN-EA-004 on the AutomationDirect Tech Notes web page.

Array Data: An array is basically a data structure of a sequential set of the same data type. Each element is accessed by an index number. Arrays can be created from Atomic data types or complex data types (pre-defined or user-defined).

All array tags are addressable in the C-more programming software. Array tags can be imported via the L5K file. You may choose to limit the number of imported elements by adjusting the "Import Array Element Count Limit" field in the C-more programming software import window.

User Defined: User-defined data types are complex data types that are created by the user. They are very much like the pre-defined structures but the user may choose which data type members the user-defined data type is composed of. It is a very efficient way to organize data in the PLC.

Here is an example of a user-defined data type:

DataType_A:

DintMember (which is a DINT member of this user-defined data type)

IntArray[10] (which is a INT array of this user-defined data type)

BoolArray[32] (which is a BOOL array of this user-defined data type)

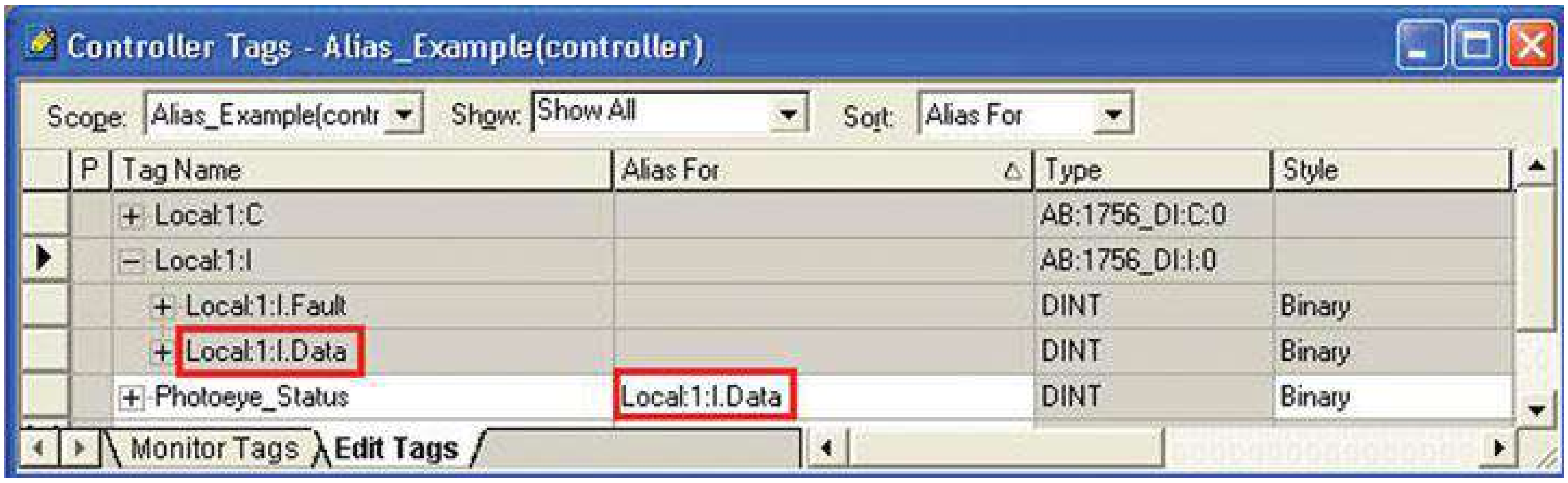
If a Tag called "Example1" was added using the "DataType_A" data type, the RS Logix5000 software would create these tags:
Example1.DintMember
Example1.IntArray[0]–Example1.IntArray[9]
Example1.BoolArray[0]–Example1.BoolArray[31]

All user-defined data types are addressable in the C-more programming software. User-defined data types are also importable via L5K file depending upon the members of the user-defined data type. Refer to application document AN-EA-004 on the AutomationDirect Tech Notes web page and see section 5.a. for details.

I/O Tag Data: I/O Tags are created automatically when an I/O module is added to the configuration. They are typically members of a complex data type. The complex data types differ based upon the module being configured and the various configuration options that the user has selected.

These data types are not directly importable into C-more via the L5K file but must be imported using the standard C-more Tag importation. There is a macro function available to help make I/O tags importation easier. I/O tags can also be manually entered.

Aliasing: Aliasing is a method to 'de-reference' the user tag from the memory location in the PLC. It could be said that Tag names themselves do this very thing but Aliasing creates another abstraction layer from tags such as I/O tags that can and will change very often. Aliasing is very useful for HMIs, where the data point is needed but the constant hassle of changing the name can become tedious. An example of an Alias in the RS Logix 5000 software is shown below:



Controller Tags - Alias_Example(controller)

Scope: Alias_Example(contr Show: Show All Sort: Alias For

P	Tag Name	Alias For	Type	Style
	+ Local:1:C		AB:1756_DI:C:0	
	− Local:1:I		AB:1756_DI:I:0	
	+ Local:1:I.Fault		DINT	Binary
	+ Local:1:I.Data		DINT	Binary
	+ Photoeye_Status	Local:1:I.Data	DINT	Binary

In the example above, “Photoeye_Status” is an Alias to the I/O tag “Local:1:I.Data”. There are several Alias import options available in C-more. Here is an explanation of these options:

“Do not import Aliases” = This means, simply, that the C-more programming software will not import any Alias tag names.

“Import all Aliases & Tags” = This means that the C-more programming software will import everything that it can. If there are many Alias tags in the project, this method will result in duplicate tag references since you will have imported the base tag itself, as well as the Alias of that tag.

“Import only Aliases” = This option will only import those Tags that have Aliases.

“Import Aliases & Tags that do not have Aliases” = This option will import Alias tags but not the base tag that the Alias refers to and it will import tags that do not have Aliases.

Unfortunately, only importation of Aliases to BOOL I/O tags can be done with this method because the memory size for all other I/O data types cannot be derived from the L5K file. To import Aliases of all non-BOOL I/O tags, use the method outlined in section 5.b. of the application document AN-EA-004 on the AutomationDirect Tech Notes web page.

Program Scope Tags: Tags can be in two locations in the PLC: Controller Scope (globally accessible anywhere) or Program Scope (only available within that program).

Program Scope tags can be accessed via C-more HMI but they require additional syntax in front of the tag name. To access a Program Scope tag, you must enter “PROGRAM:program name.tag” name where PROGRAM is static text to identify that a Program Scope tag is following.

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

The C-more programming software supports the import of tag database files that are stored in a tabular format, also known as a ".csv" file type. For a detailed example, we will discuss importing from an existing L5K file which is created with the RSLogix 5000 programming software.

Porting Logix Tags Into C-more Programming Software

The RSLogix 5000 software allows users to create an export file that contains the Tag database information. Within RSLogix 5000 and with your project file open, Select File > Save As (Figure 1).

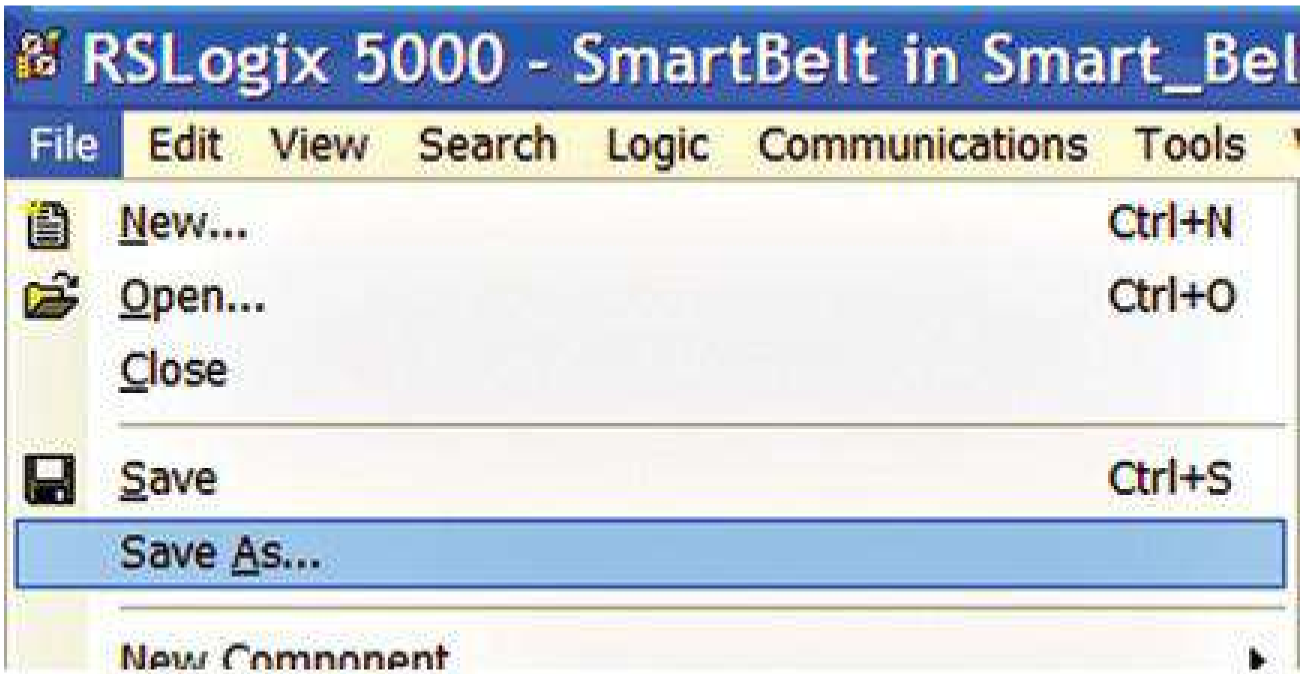


Figure 1

Then change the "Save as type" to L5K (Figure 3). Remember which folder you selected when saving the file since you will need to find it later with the C-more programming software.

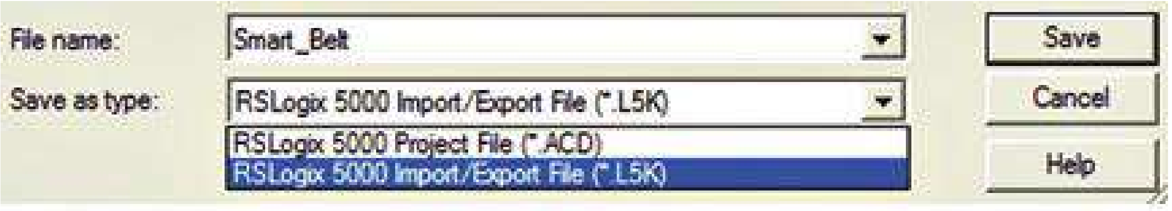

File name: Smart_Belt
Save as type: RSLogix 5000 Import/Export File (*.L5K)
RSLogix 5000 Project File (*.ACD)
RSLogix 5000 Import/Export File (*.L5K)
Save
Cancel
Help

Figure 3

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Now that you have the L5K File saved, open the C-more Programming Software and select Import > Tag Name Database (Figure 4).

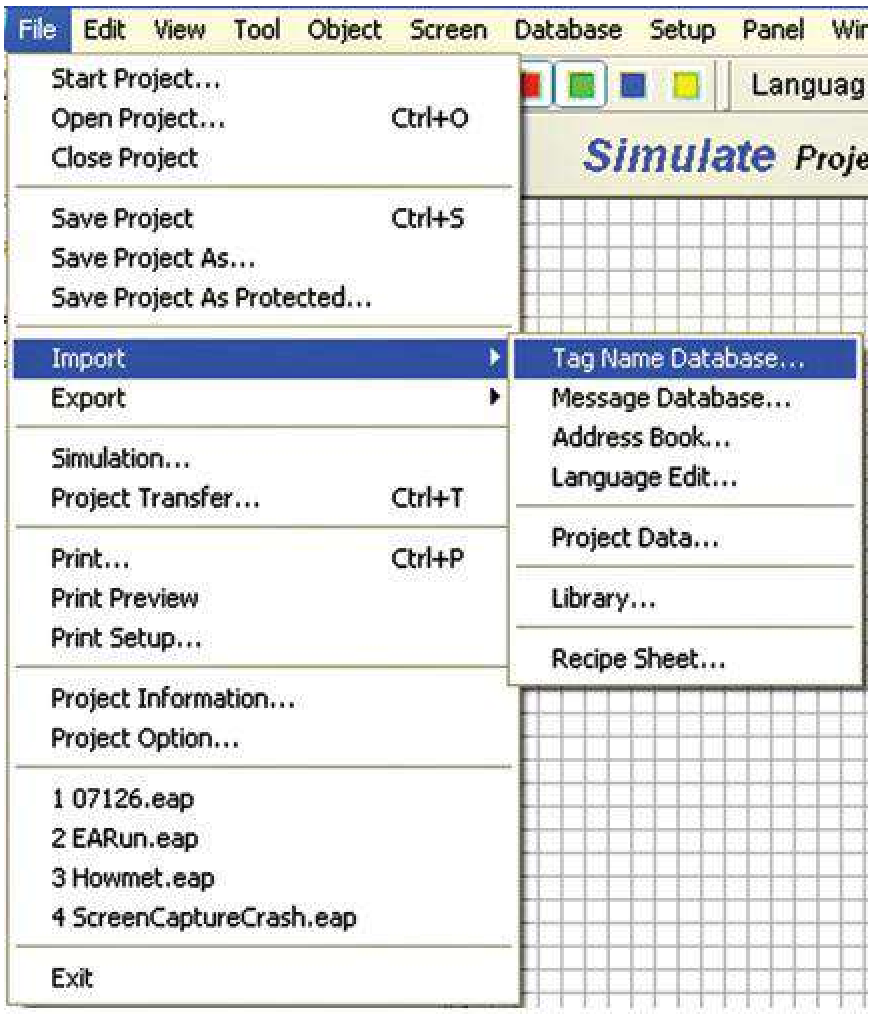
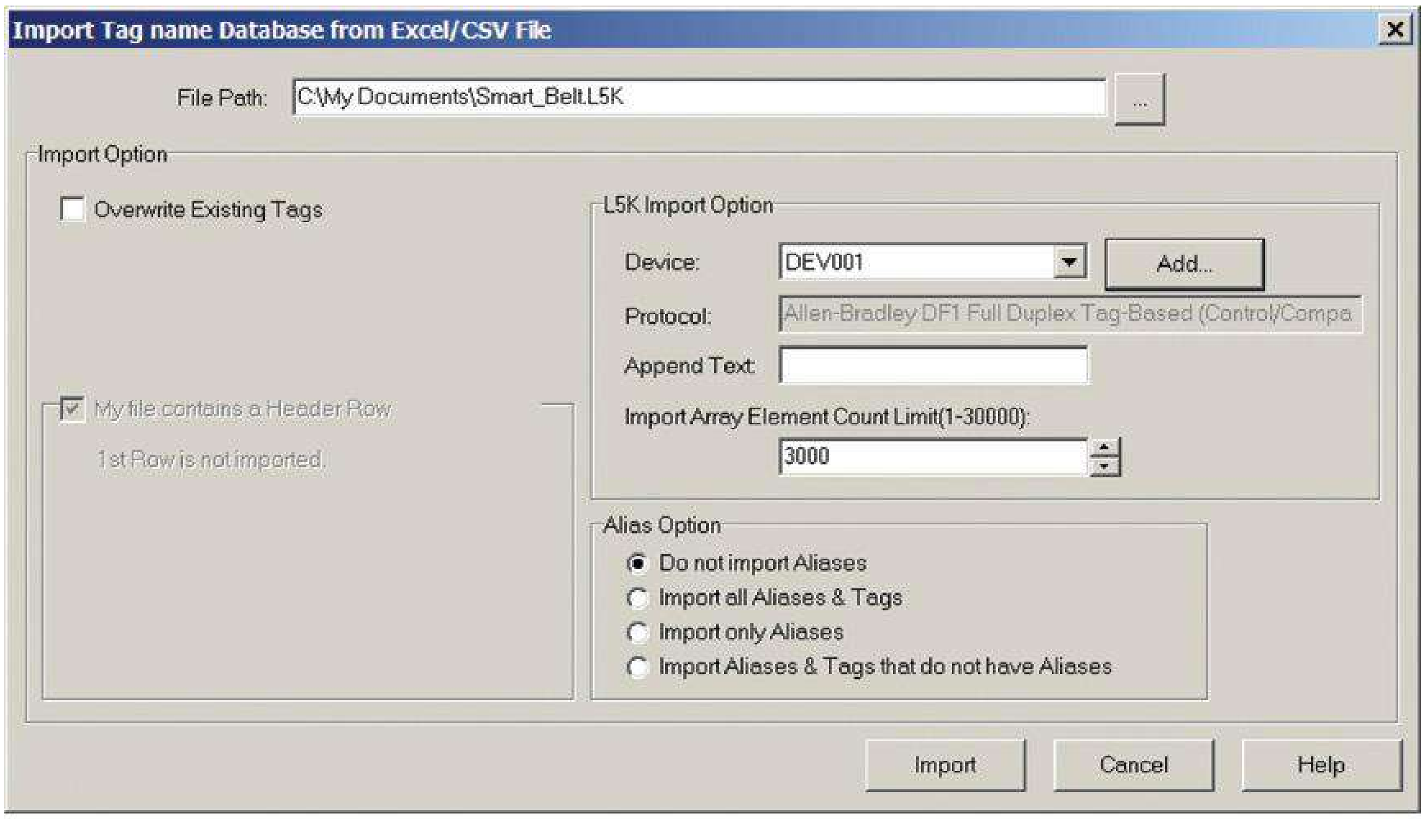


Figure 4

This will open a window that will display options for importing the tag database files. (Figure 5)



Import Tag name Database from Excel/CSV File

File Path: C:\My Documents\Smart_Belt.L5K

Import Option

☐ Overwrite Existing Tags

☑ My file contains a Header Row

1st Row is not imported.

L5K Import Option

Device: DEV001

Protocol: Allen-Bradley DF1 Full Duplex Tag-Based (Control/Compa

Append Text:

Import Array Element Count Limit(1-30000): 3000

Alias Option

● Do not import Aliases
○ Import all Aliases & Tags
○ Import only Aliases
○ Import Aliases & Tags that do not have Aliases

Figure 5

Jump to Chapter
Chapter 1 What is an HMI and What Can It Do for You?
Chapter 2 Good HMI Software: What Capabilities to Look for
Chapter 3 Pick and Place Your HMI
Chapter 4 16 Tips for an Effective HMI
Chapter 5 Remote Monitoring with Embedded HMIs
Chapter 6 Layered Security Goes Beyond Hardware
Chapter 7 System Integrators Talk HMI Projects
Chapter 8 Working with Objects
Chapter 9 Events and Event Actions
Chapter 10 Understanding Recipes
Chapter 11 Data Logging and Trends
Chapter 12 Project Simulation
Chapter 13 Tutorial: Simple AC Drive Control Interface
Chapter 14 Using C-more with Allen-Bradley PLCs

Importing Predefined Data Types

There are many pre-defined data types in the A-B ControlLogix and CompactLogix PLCs. A pre-defined data type is one that is already created in the RS Logix 5000 software when starting a new project. Some common pre-defined data types are TIMER and COUNTER. The complication in importing pre-defined data types is that there are many of these data types in the RSLogix 5000 software and within many of those, there are multiple data type members. Many members of these pre-defined data types, while useful in the PLC application, will not be needed in the C-more project. So, the issue is to decide which pre-defined data types should be imported and which members of those types should or shouldn't be imported.

Optimizing Communications Between C-more and A-B PLCs

In most situations, specific actions to optimize the communications between C-more and the PLC are not necessary. In a large application where performance is crucial, it may require more careful planning when designing the system. The following guidelines can be used to "optimize communications with most types of HMI brands connected to a PLC. However, when using a C-more HMI connected to a PLC, some detailed knowledge of how and when communications are initiated is necessary.

Writing Data to the PLC

C-more writes data to the PLC infrequently in most applications.

Screen objects only write to the PLC when triggered by the user, such as when a Pushbutton object is pressed or released, the Enter key is pressed in a Numeric Entry object, the Slider object is moved, etc.

The second way that C-more will issue a Write action to the PLC is through the Event Manager's "Tag Copy" function. This can be triggered by a tag changing values, a screen change or time schedule. (See Event and Event Actions Chapter for more details.)

The third way that C-more will write to the PLC is through the Panel to PLC tags. These typically only occur on a value change and happen infrequently most of the time.

Writes occur rarely enough in most cases and typically do not need to be considered when trying to optimize communications.

Reading Data From The PLC

There are several functions which cause C-more to issue read requests to the PLC.

The most common are the objects on the current screen that is being displayed. The only objects that will issue read requests, regardless of the screen being displayed, are the trend objects. Most of the time, optimizing the PLC addresses assigned to objects on the same screen will have the most impact on performance.

Additional HMI functions that will issue Read requests are alarm events and tag events. Any event created with a tag object addressed to the PLC will constantly read from the PLC no matter which screen is being displayed. Careful planning of the addresses used in events will greatly increase communications performance as well.

The PLC to Panel tags will also be read constantly regardless of which screen is being displayed. Logical PLC addressing of these tags should also be considered.

Optimization for Non-Tag Based PLCs (SLC and MicroLogix)

The SLC and MicroLogix PLCs are fixed addressed PLCs. Grouping consecutive addresses of the same data type in the same file without significant gaps between the addresses is the only way to optimize communications for these PLCs. Grouping tags of consecutive addresses on the same screen, in the event manager or in the PLC to Panel screen will reduce the number of requested packets and will therefore increase the frequency of updates to the visible objects.

Optimization for Tag Based PLCs (Control/Compact/FlexLogix PLCs)

The Control/Compact and FlexLogix PLCs are tag based PLCs. The method of communications (both serial and Ethernet) involves including the actual symbolic tag name in the request sent over the wire. This means the tag names themselves affect the speed of communications by creating larger and/or multiple requests to the PLC. Program space tag names are not efficient; each tag name needs the PROGRAM: header as well as the name of the program in front of the tag name. This scheme makes the size of the request much larger, requiring more requests to get multiple tags.

Controller scope tags are globally accessible and do not require the header information, thus making the size of the request much smaller and more efficient. Controller Scope tags should be used as much as possible when performance optimization is an issue.

Another method to reduce the size of the tag name requested without necessarily compromising the descriptive nature of the name is to use an Alias name in the C-more software. The alias name could be much shorter than the original tag name in the PLC.

The most effective method to reduce the number of requests from C-more to the PLC is to use arrays in the PLC and then assign the C-more tags to consecutive elements in the PLC array. C-more can then request multiple PLC tags with one request. This method will, by far, have the most impact in enhanced communication performance between C-more and the PLC.

Step-by-step instructional video on importing Allen Bradley Tags into your C-more HMI:

Learn More...Watch The Video!

C-more HMI Setup: Import Micro800 PLC Tags

http://n2adc.com/z6ksr

Learn More...Watch The Video!

C-more HMI Setup: Import RSlogix 500 PLC Tags

http://n2adc.com/i65gp

Learn More...Watch The Video!

C-more HMI Setup: Import Studio 5000 Tags

http://n2adc.com/h4ft7