

HMI HANDBOOK



Practical Guide to HMI Selection, Design and Operation

HMI Handbook

TABLE OF CONTENTS

General HMI

Chapter 1

What is an HMI and What Can It Do for You?3

Chapter 2

Good HMI Software: What Capabilities to Look For6

Chapter 3

Pick and Place Your HMI 13

Chapter 4

16 Tips for an Effective HMI 16

Chapter 5

Remote Monitoring with Embedded HMIs 21

Chapter 6

Layered Security Goes Beyond Hardware..... 26

Chapter 7

System Integrators Talk HMI Projects 30

C-more Specific

Chapter 8

Working with Objects..... 33

Chapter 9

Events and Event Actions..... 40

Chapter 10

Understanding Recipes 43

Chapter 11

Data Logging and Trends..... 48

Chapter 12

Project Simulation 55

Chapter 13

Tutorial: Simple AC Drive Control Interface..... 58

Chapter 14

Using C-more with Allen-Bradley PLCs 72

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 let's 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.

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

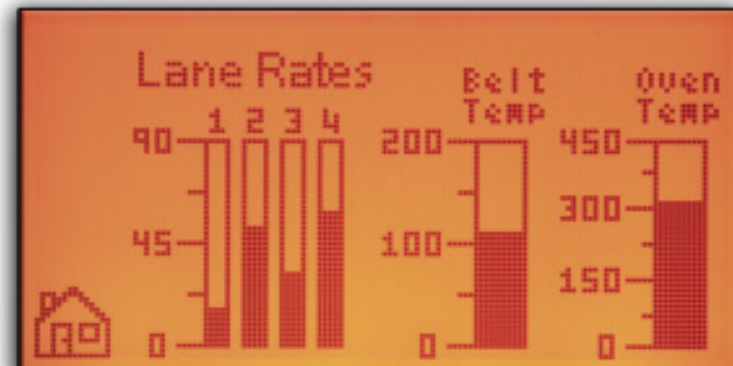
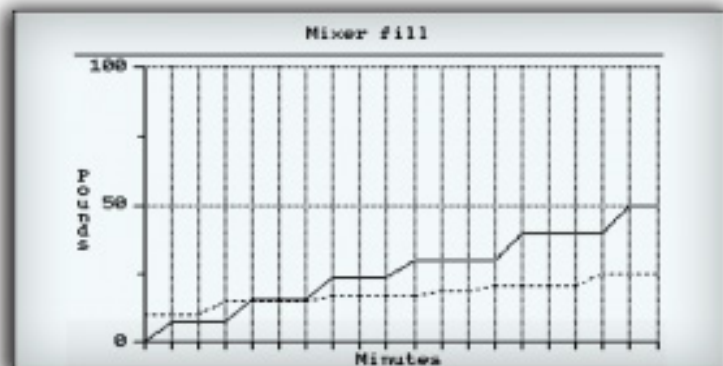
Chapter 14

Using C-more with
Allen-Bradley PLCs

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

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.



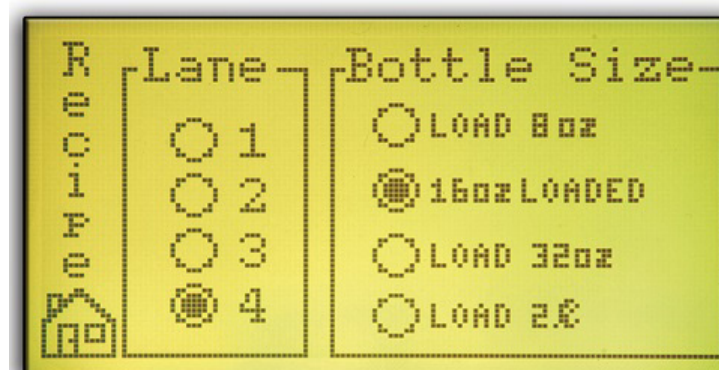
Data Entry – Easily change process variables with a pop-up numeric keypad on the screen or with increment/decrement arrows.



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 1

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



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:

$\$171^* \text{ (beacons)} \text{ vs } \$98^* \text{ (HMI)} + \$69^* \text{ (CLICK PLC)} = \$167^* \text{ (PLC and HMI)}$

"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.

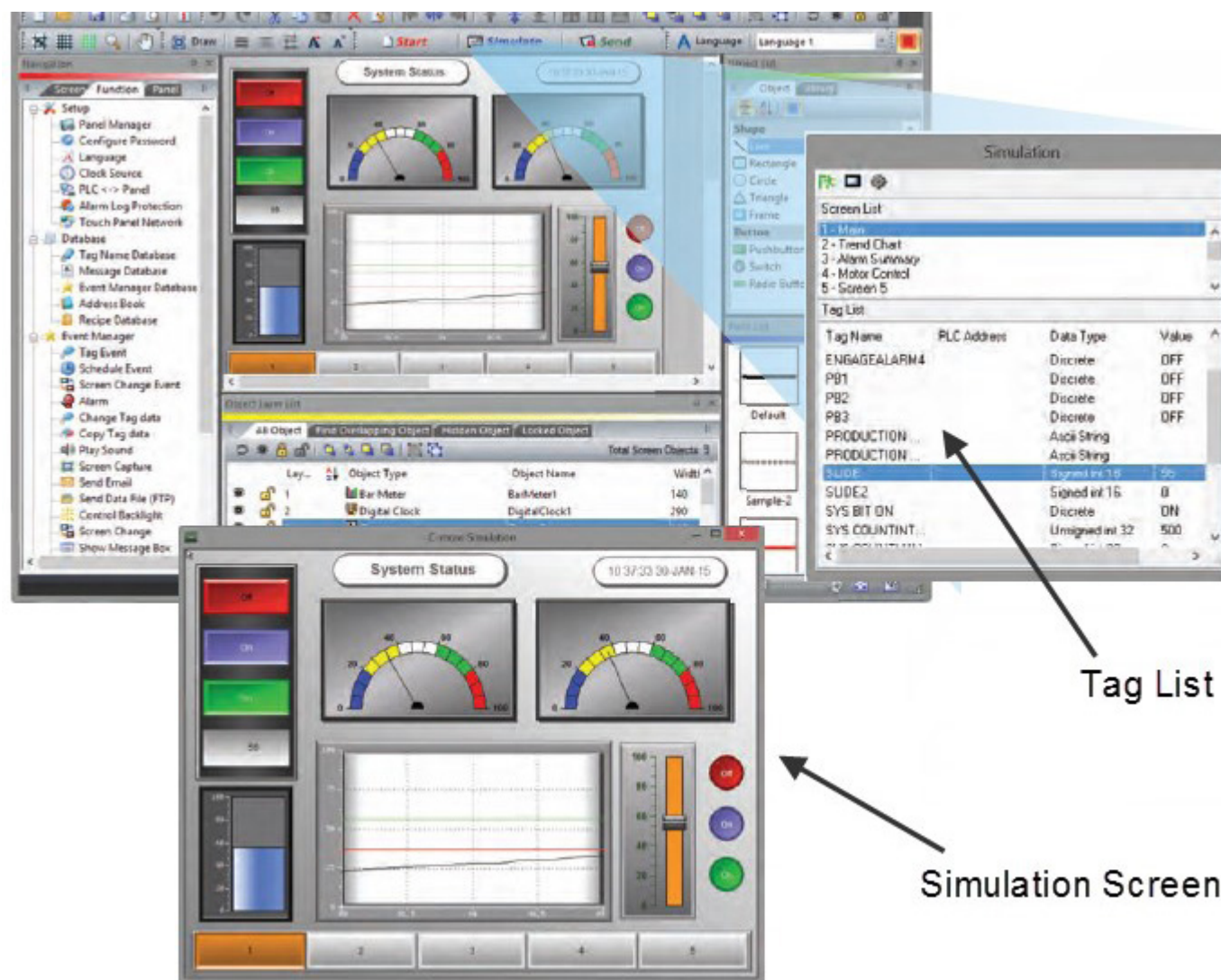
In any project, cost is a primary concern. For simple systems, micro HMIs offer many benefits over mechanical pilot devices including more functionality, time-saving modifications and overall cost. So if you're thinking the budget or the system itself is just too small for an HMI, take a look at the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

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

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...



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 from 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

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.

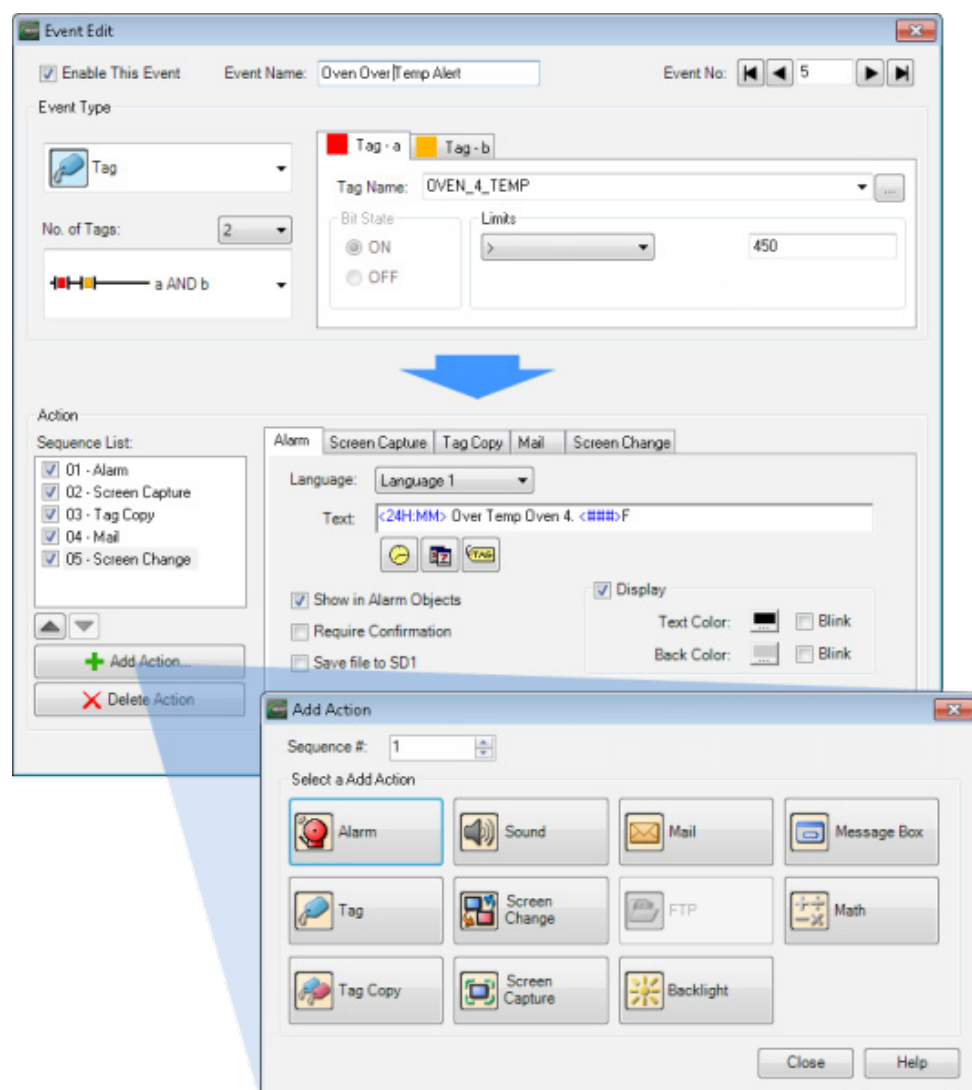
The operator should be able to modify and save recipes on-the-fly while the machine is running (given that the operator has the appropriate level of access). You should also be able to quickly import and export your recipe data to a spreadsheet for further manipulation or archival purposes.



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.



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

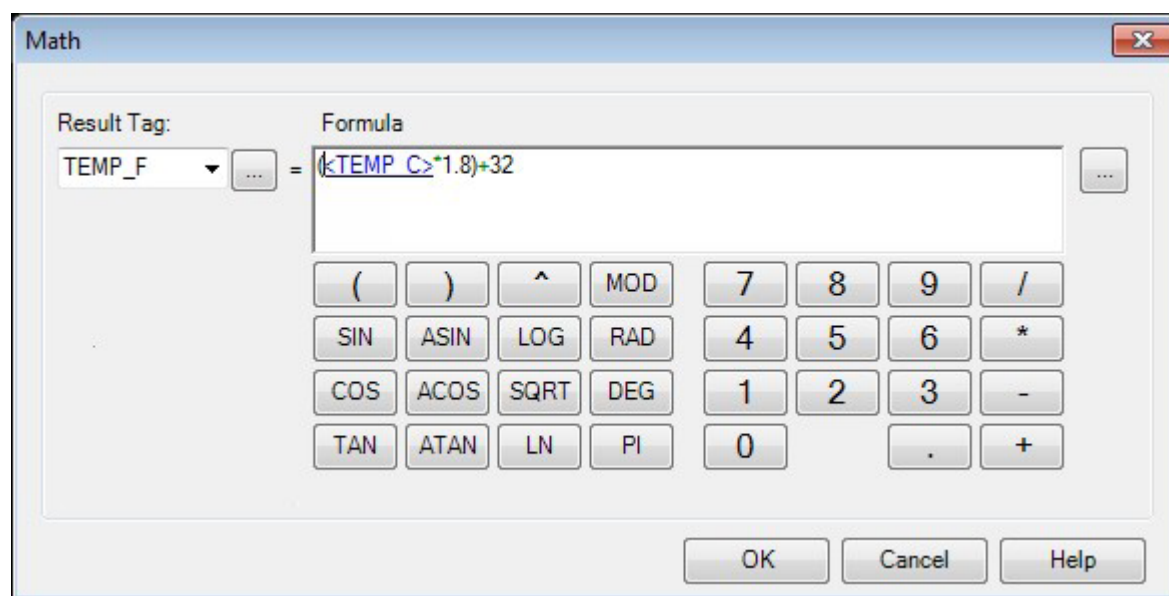
Chapter 14
Using C-more with Allen-Bradley PLCs

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...



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

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 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

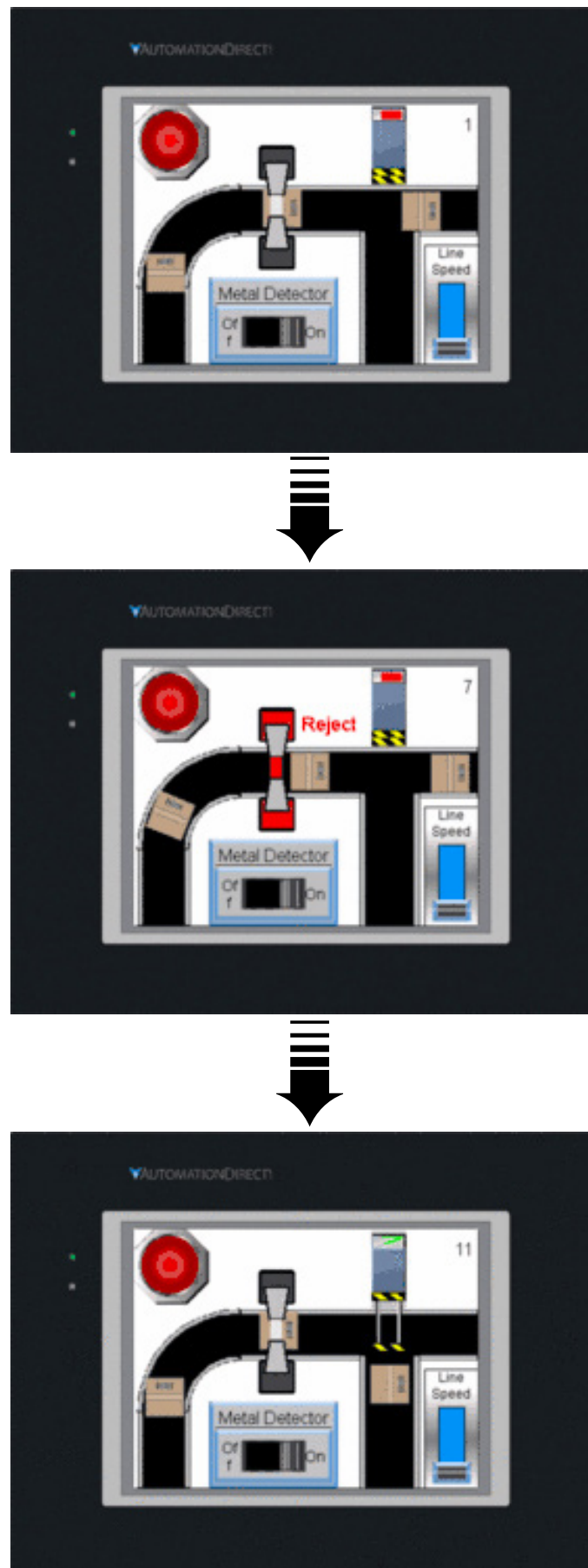
Chapter 11
Data Logging
and Trends

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 13Tutorial: Simple AC
Drive Control Interface**Chapter 14**Using C-more with
Allen-Bradley PLCs

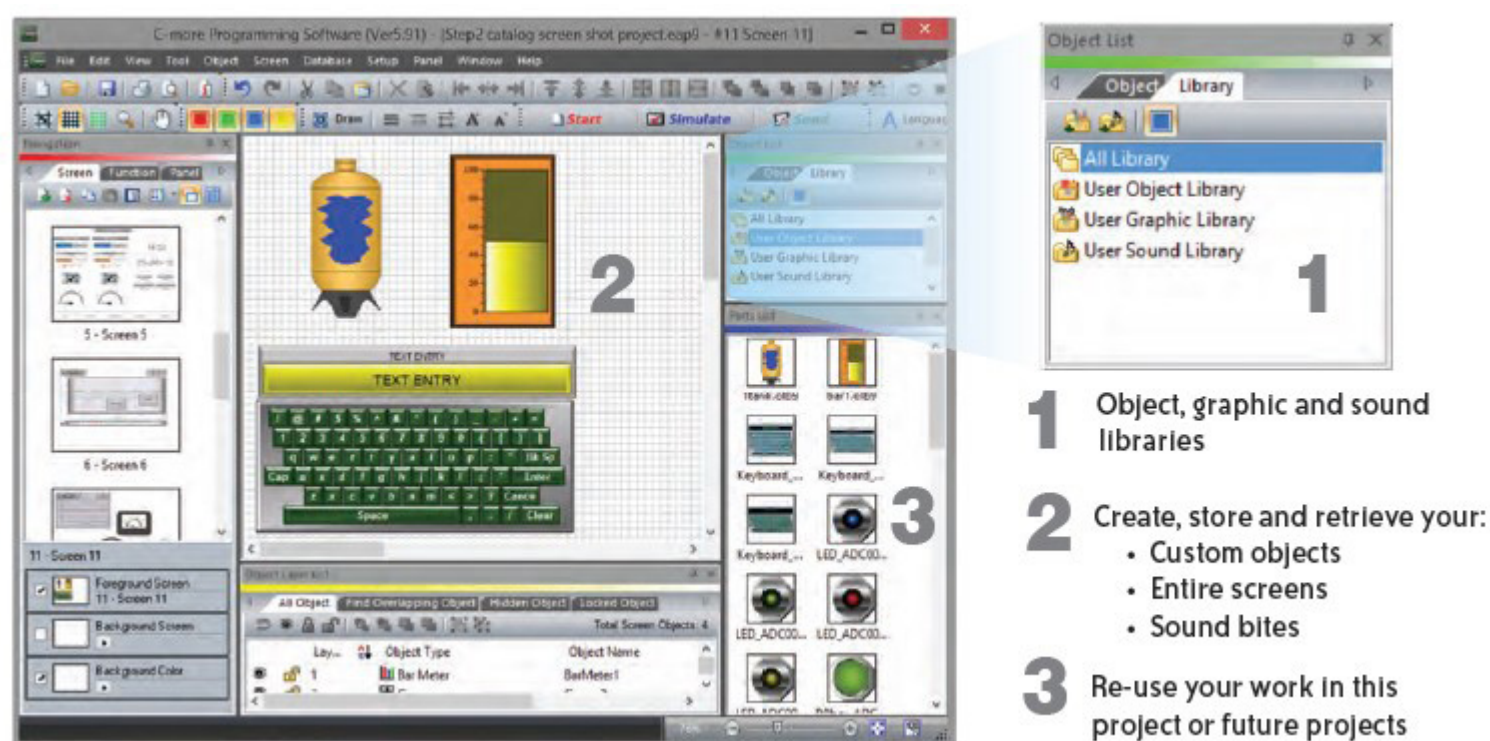
Good HMI software should make animation easy. Animations of machine motion and factory processes can enhance comprehension, and give operators a visual representation of the desired motion of products, and machinery on screen. In fault conditions, operators can quickly see the relationship of product 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.



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.



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

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

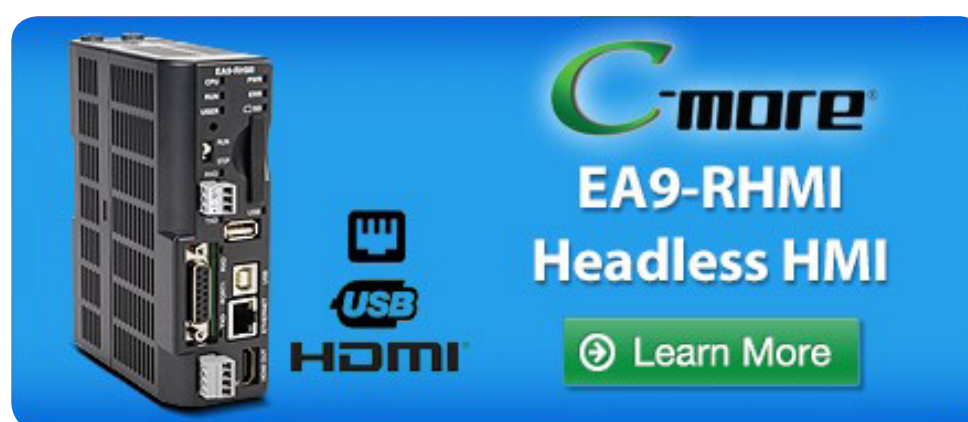
A big consideration is where and how to position the HMIs. It is very important to consider ergonomic and normal operator position. If the operator has a kink in his neck from looking at an awkwardly positioned HMI screen, he or she may start to ignore the screens and follow gut instinct. In general, 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.



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

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.

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

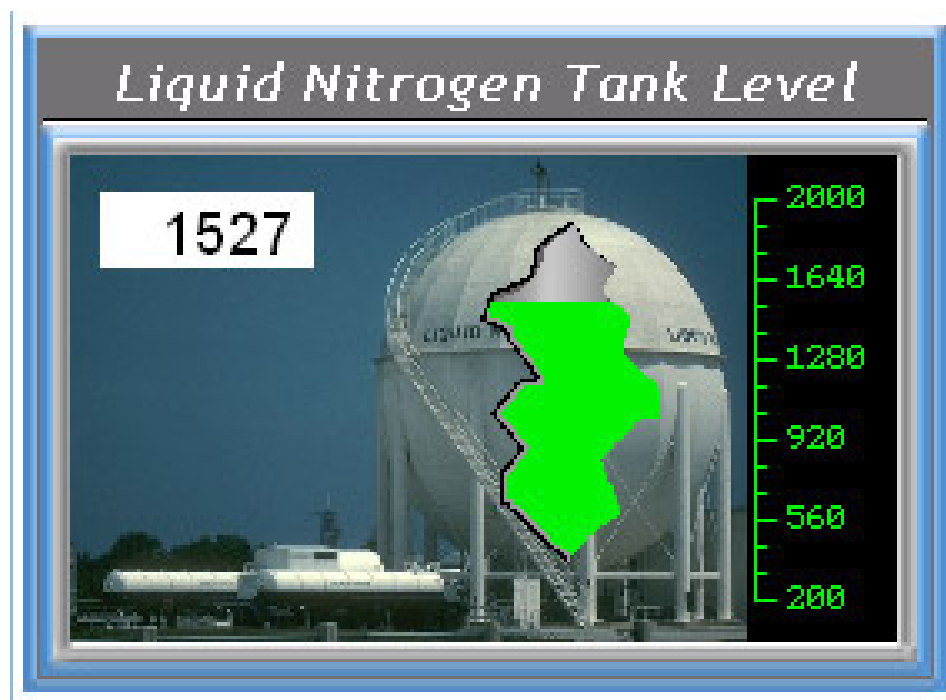
Chapter 12
Project Simulation

Chapter 13
Tutorial: Simple AC
Drive Control Interface

Chapter 14
Using C-more with
Allen-Bradley PLCs

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.



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

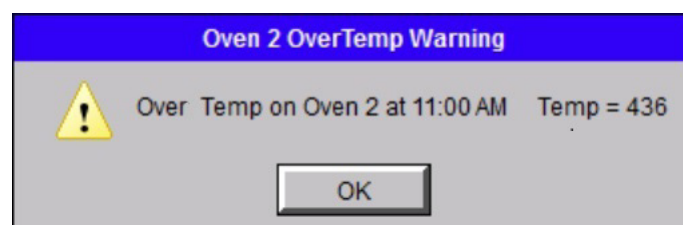
Give the operator visual (and audible) clues that buttons have been pressed, or that certain steps have already been performed. Affirmative feedback can build confidence, and satisfaction with the system. Use color and/or animation to lead operators step-by-step through complicated processes if necessary. 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.



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



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

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



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

Learn More...Watch The Video!

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



<http://n2adc.com/6qkdI>

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.

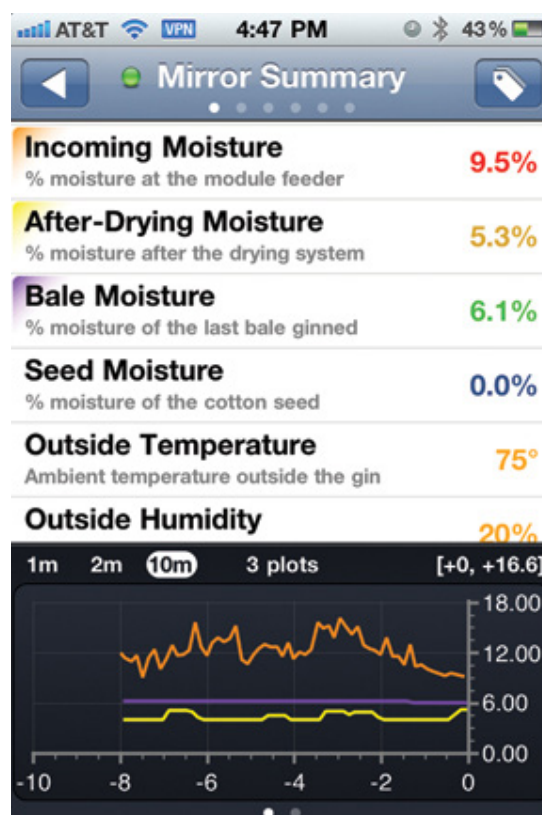


Security

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.



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 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

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 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



<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.

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

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.

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

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



Layered Security Goes Beyond Hardware

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.



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 routes data between two devices, often over unknown networks, it opens a port to a hacker, so data could be intercepted and possibly changed.

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 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.

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

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.

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

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.

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

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 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.

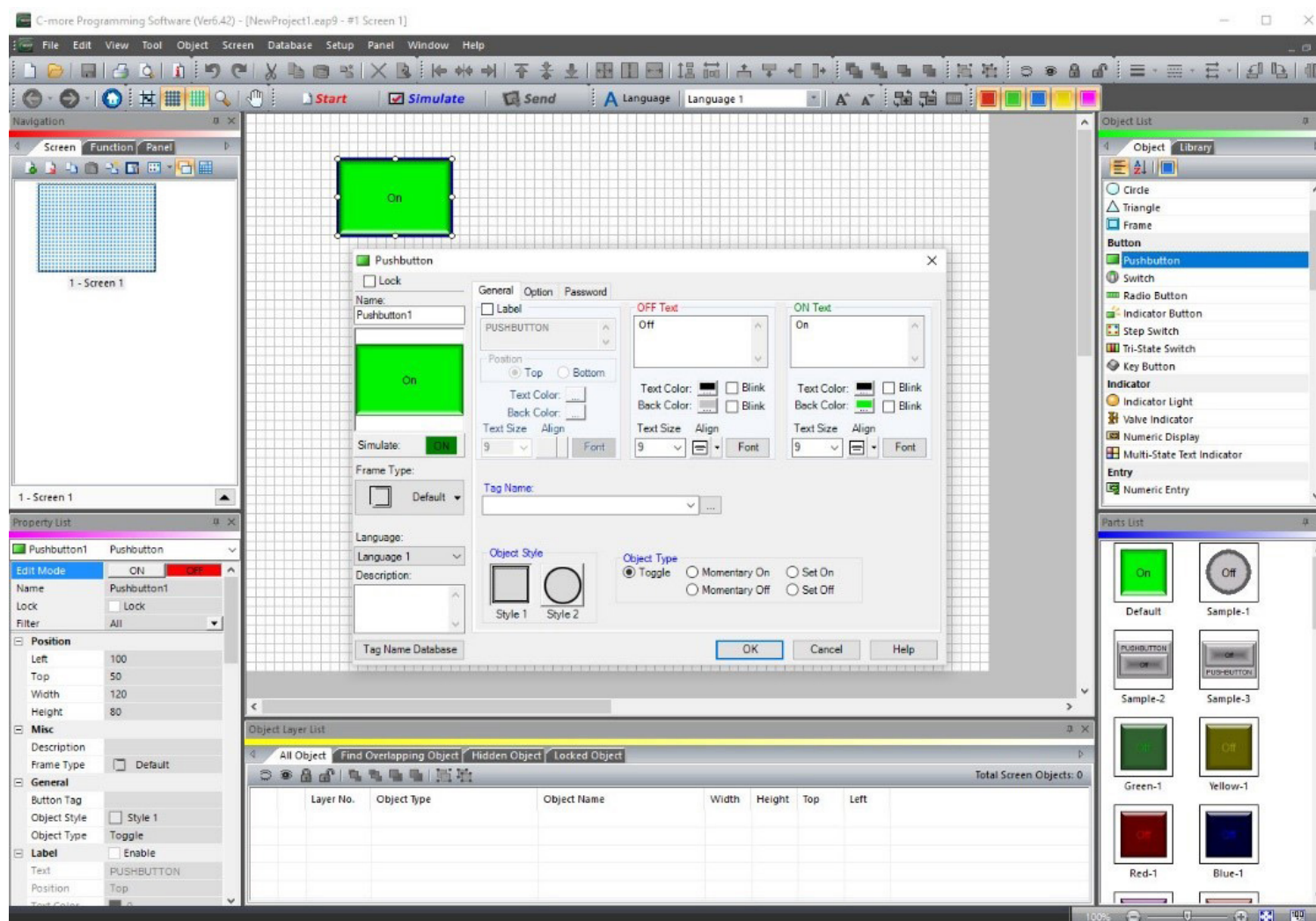
Objects available include:

- 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.



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

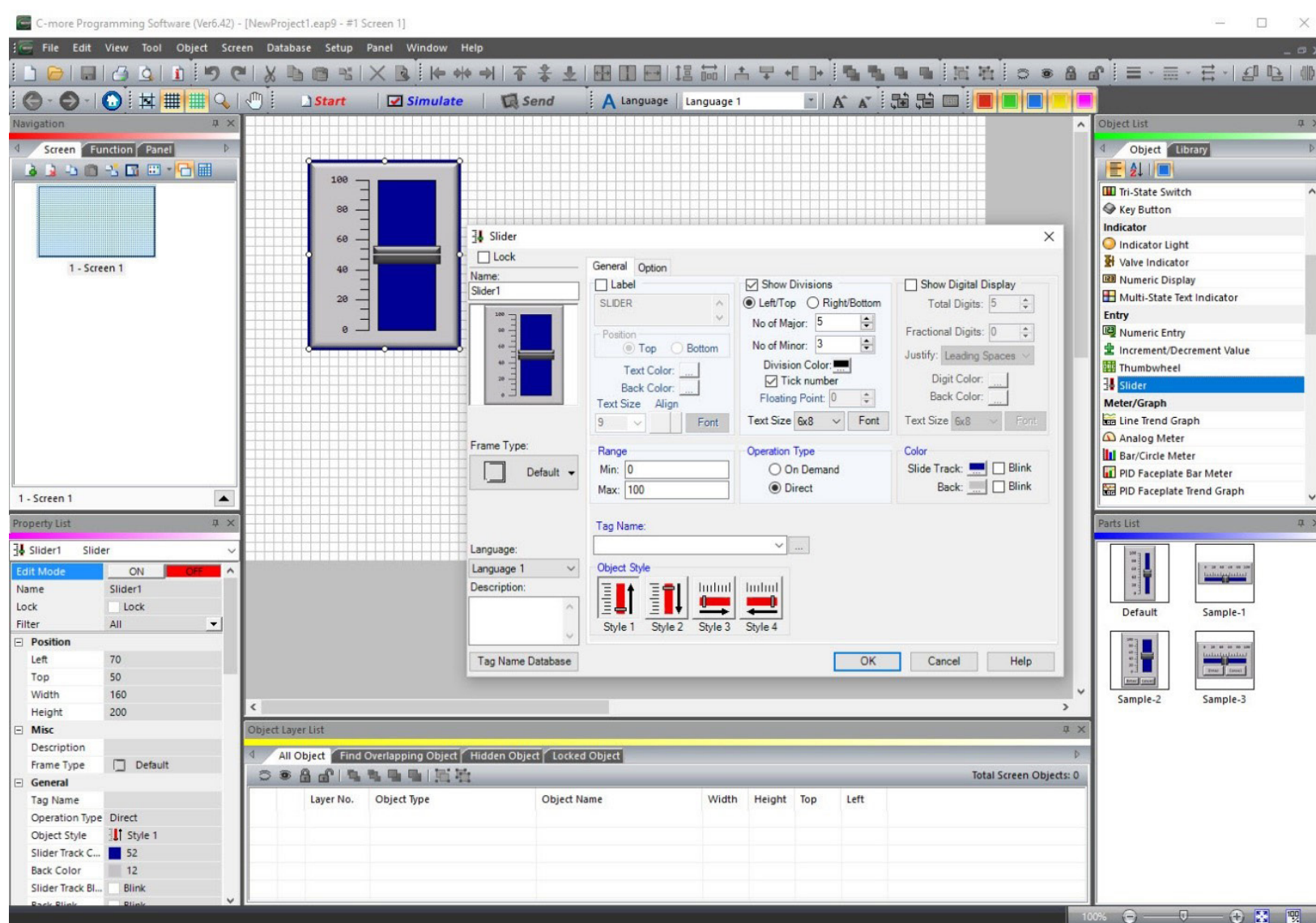
Chapter 11
Data Logging
and Trends

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.



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

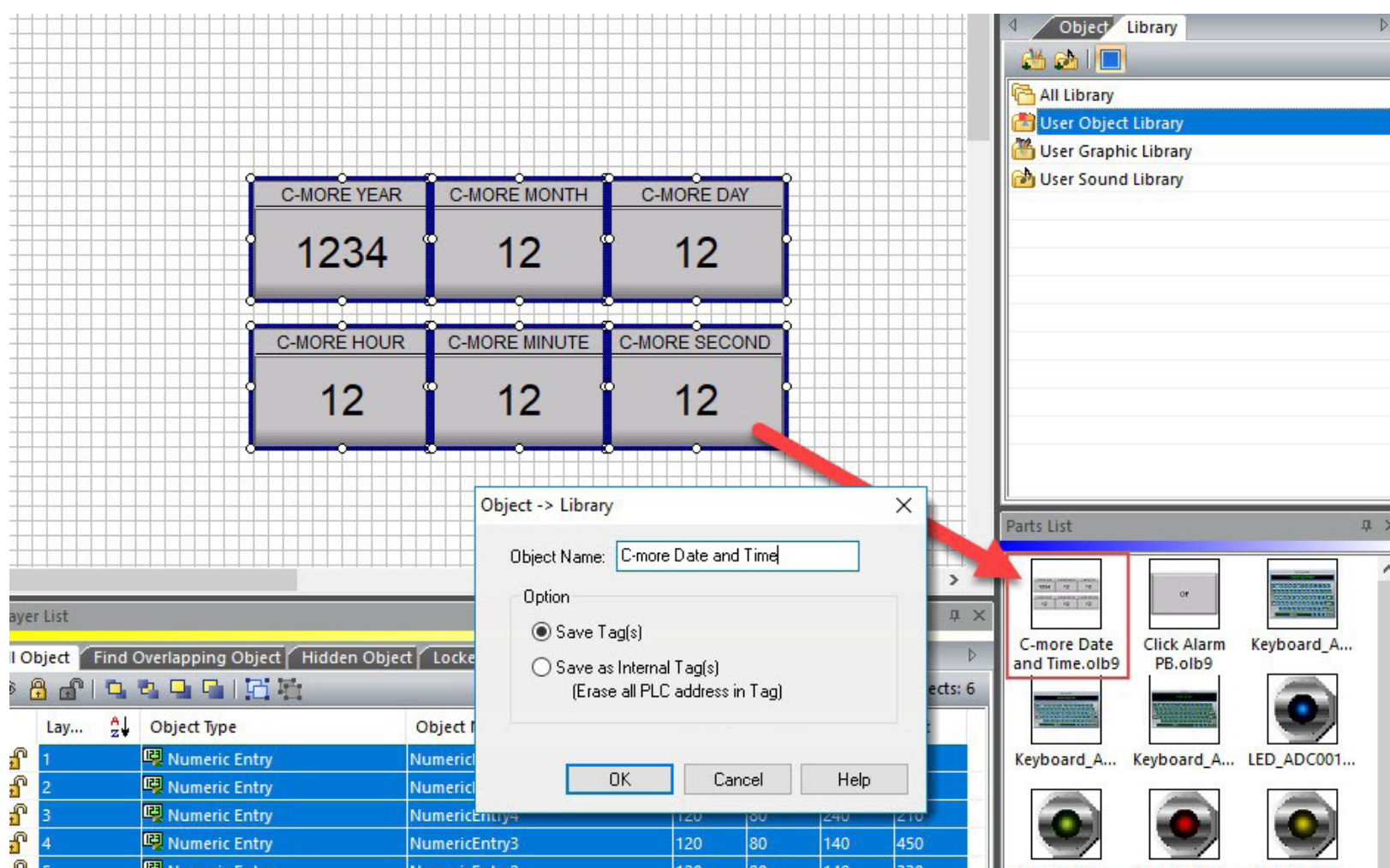
Chapter 12
Project Simulation

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.



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

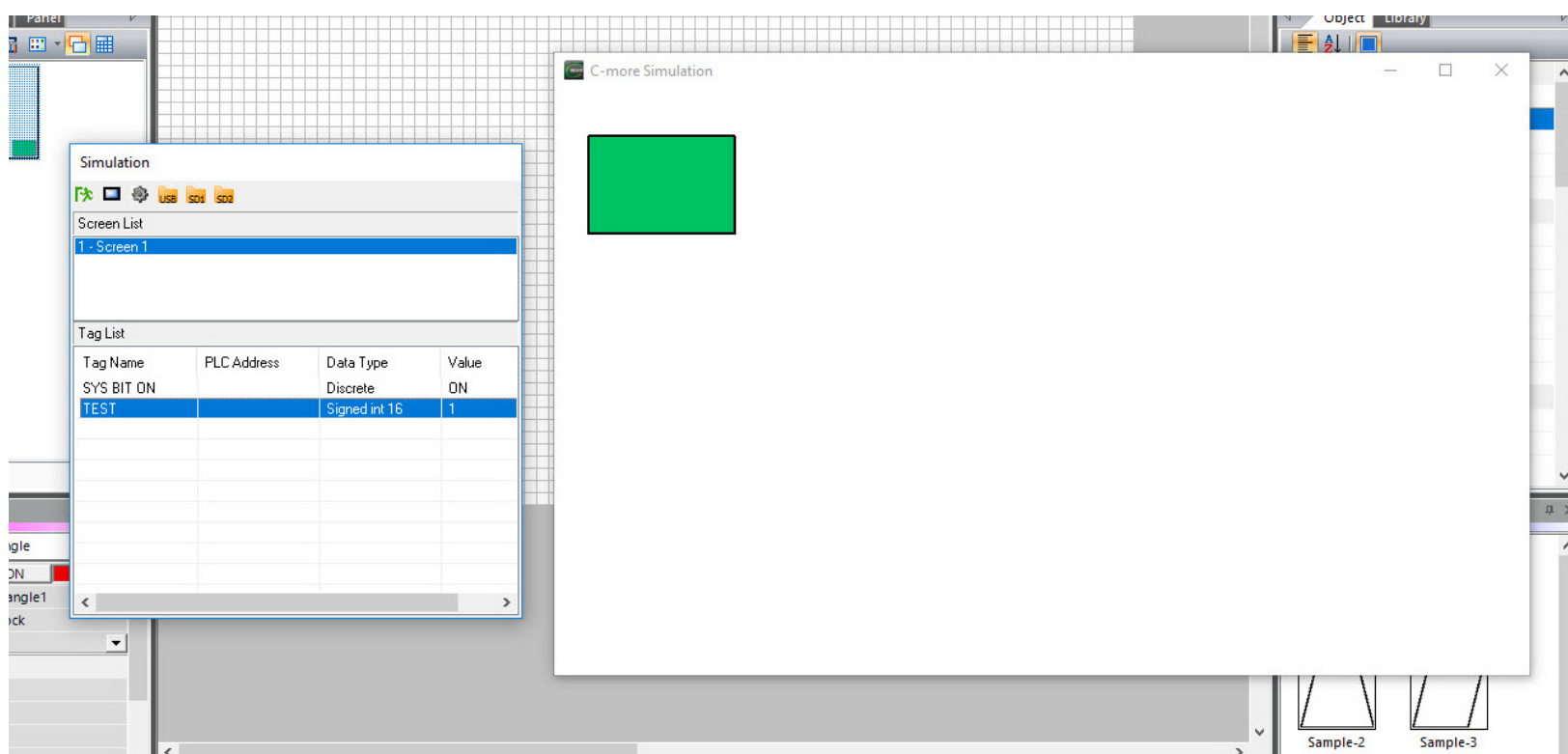
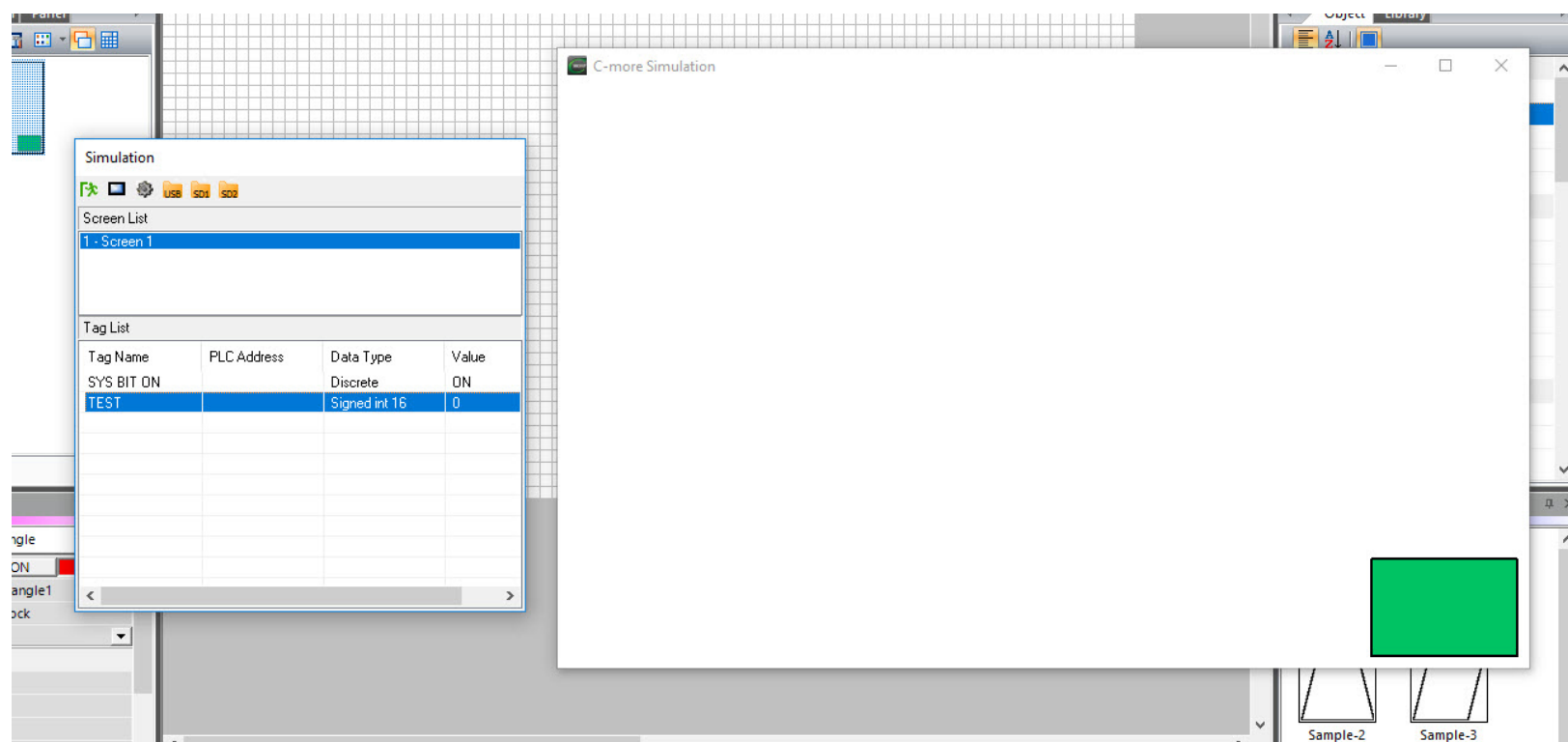
Chapter 14
Using C-more with Allen-Bradley PLCs

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)



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

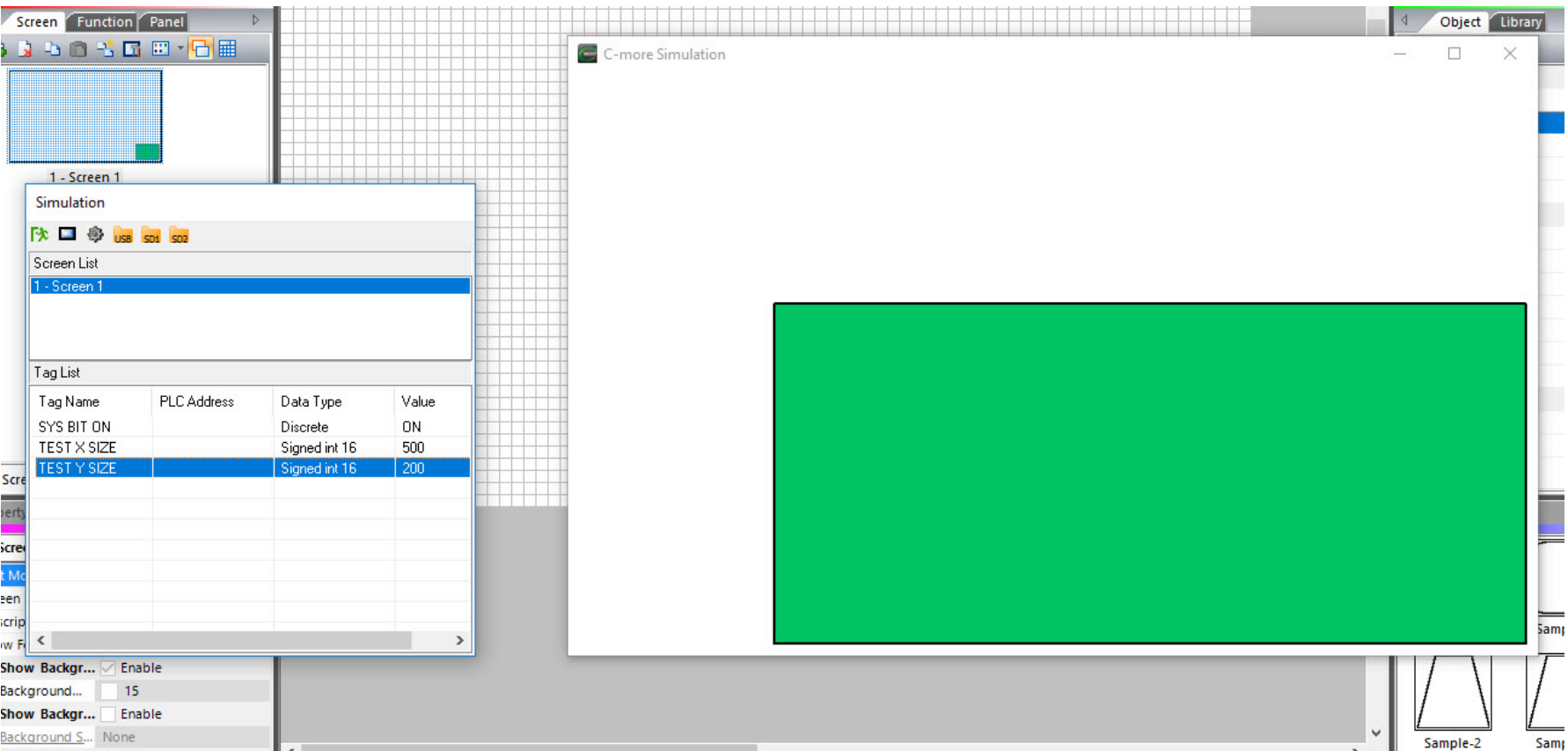
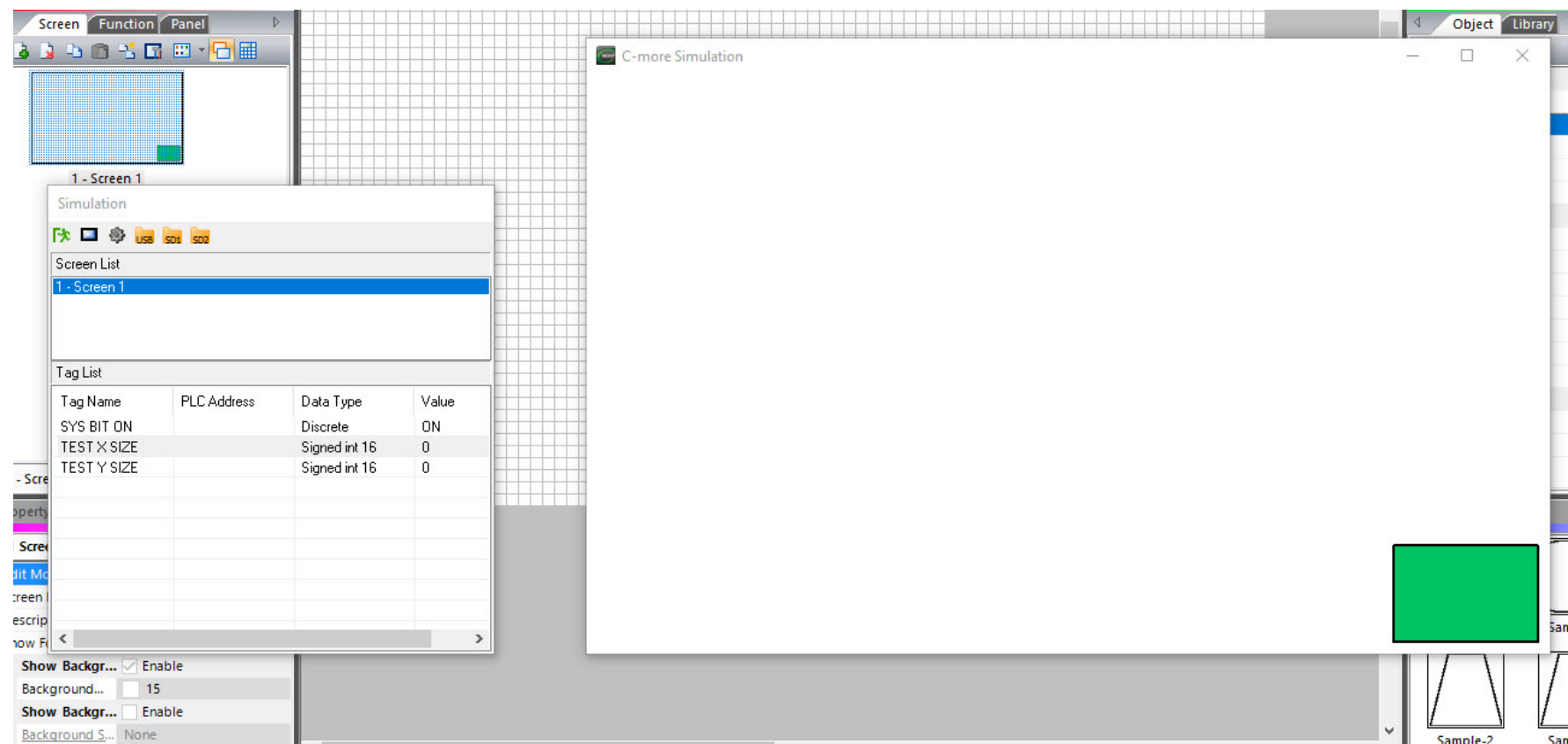
Chapter 11
Data Logging and Trends

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



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

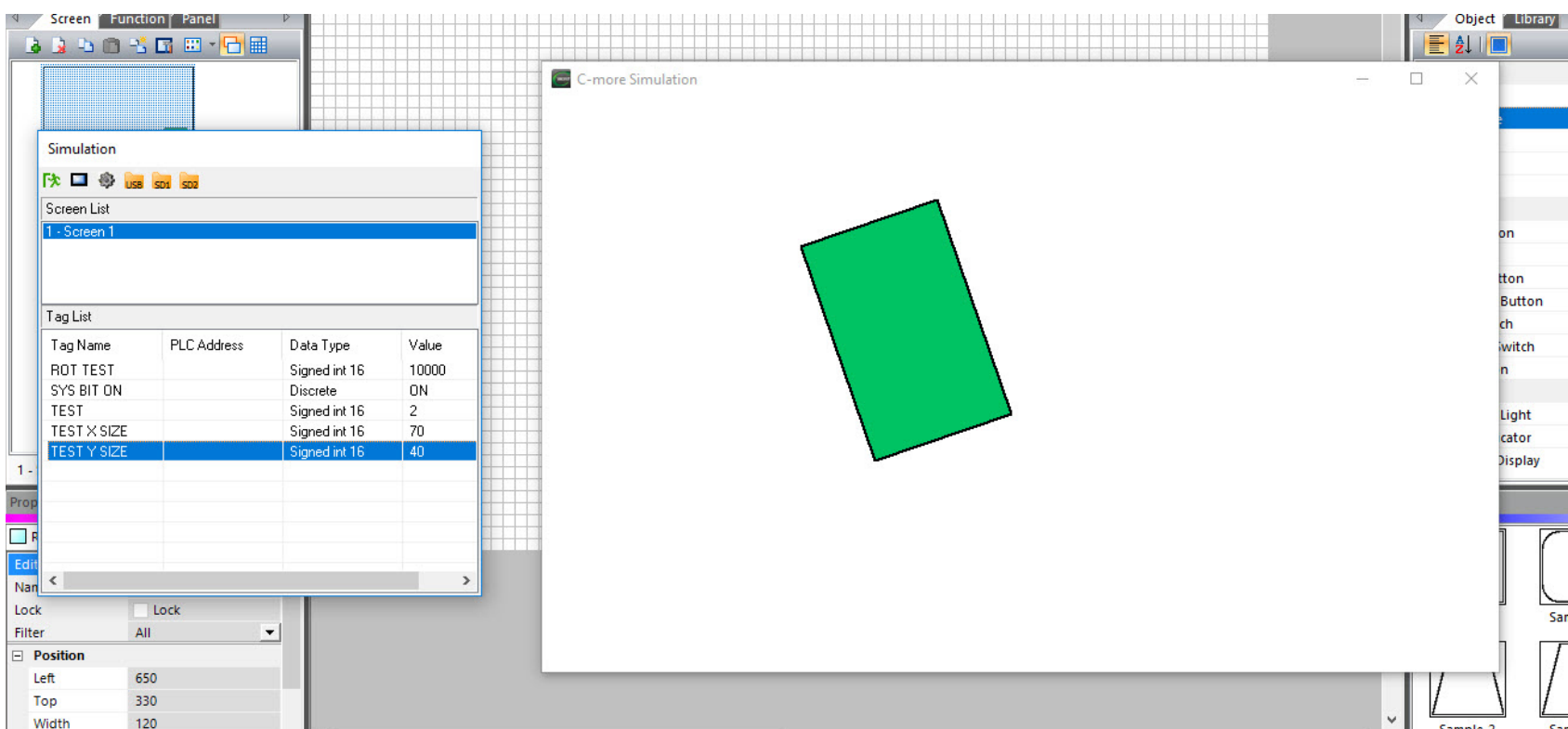
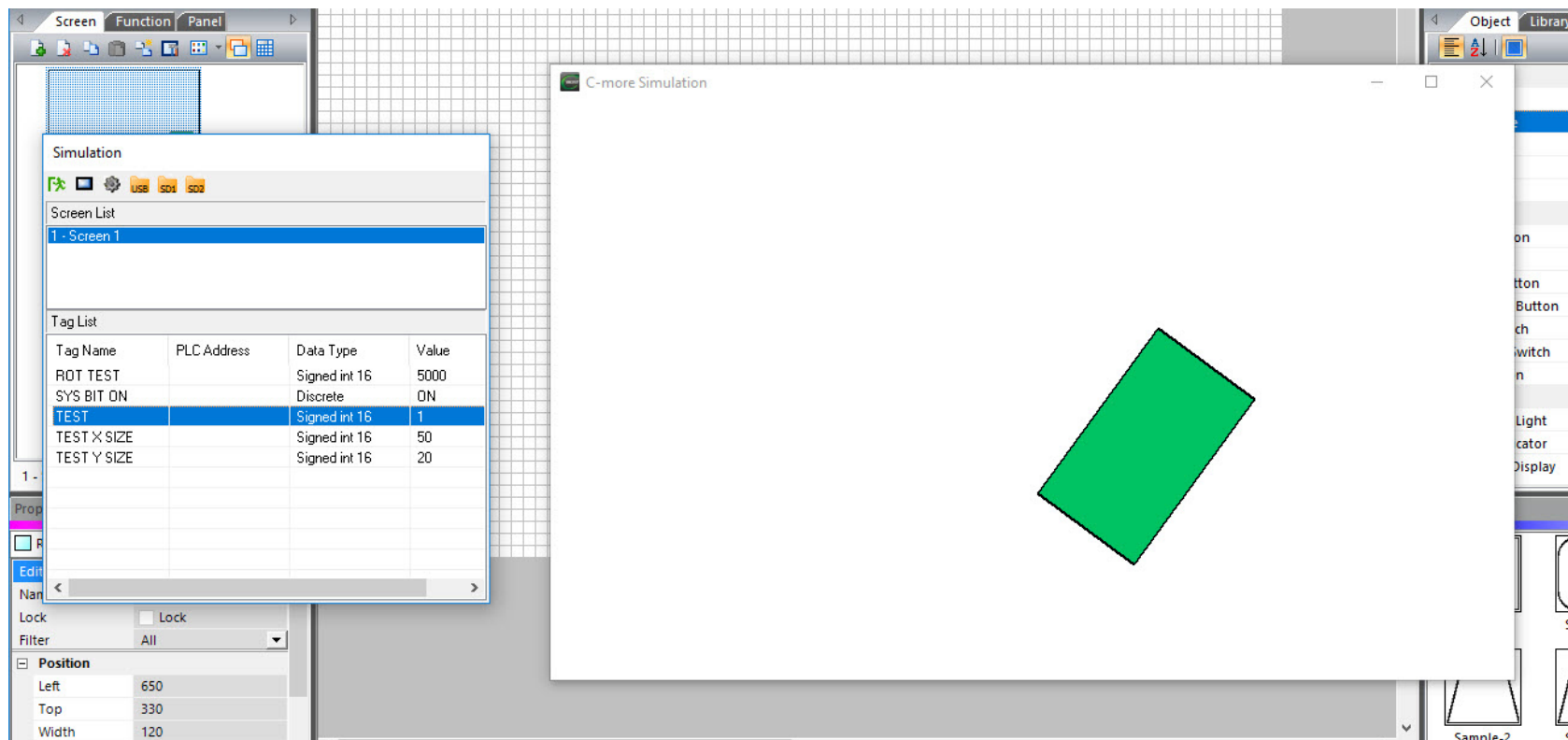
Chapter 11
Data Logging and Trends

Chapter 12
Project Simulation

Chapter 13
Tutorial: Simple AC Drive Control Interface

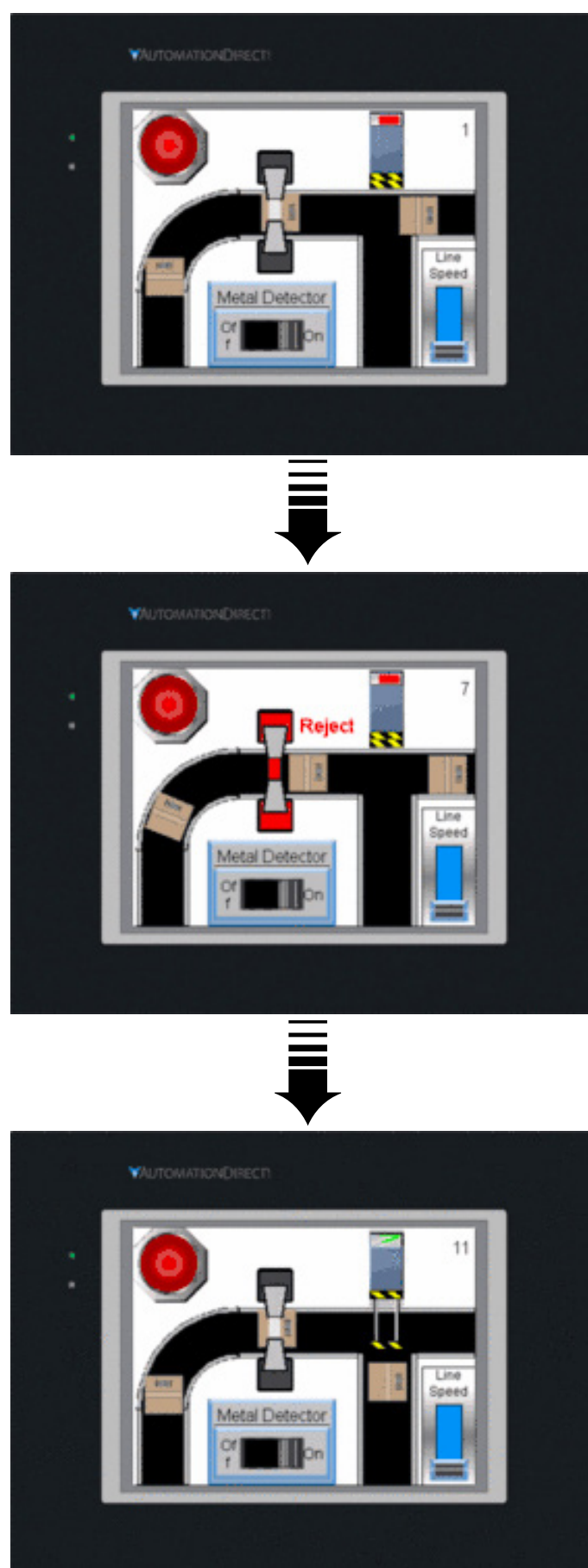
Chapter 14
Using C-more with Allen-Bradley PLCs

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



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

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.

The screenshot displays the 'Event Edit' window for an event named 'Weekly Qty Shipped Report'. The event is enabled and set to trigger on a 'Day & Time' basis every Thursday at 19:00. The event number is 1. The action sequence consists of two steps: '01 - Screen Capture' and '02 - Mail'. The 'Screen Capture' action is configured to capture 'Other Screen' (Screen No. 1, 'Loading Belts') and save it as 'Weekly Qty Shipped.jpg' in the 'Day Shift' folder. The 'Mail' action is configured to send an email to 'DayShiftSup' with the subject 'Weekly Qty Shipped' and a message containing a screen capture and quantity shipped information. The email is attached as 'Event#1 - Action#1 - [#1 Lo]'. The window includes 'Apply', 'Close', and 'Help' buttons.

Example 1

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.

The figure shows four sequential screenshots of the 'Event Add' dialog box, illustrating the configuration steps for an Estop event. Blue arrows indicate the flow from one step to the next.

Step 1: Event Setup
 - **Enable This Event:** ☒
 - **Event Name:** Estop LB1
 - **Event No.:** 1
 - **Event Type:** Tag
 - **Tag Name:** ESTOP LOADING BELT1
 - **Bit State:** ☒ OFF
 - **Limits:** Outside Range, Min: 0, Max: 4294967295

Step 2: Alarm Action Configuration
 - **Sequence List:** 01 - Alarm, 02 - Message Box, 03 - Screen Change, 04 - Sound
 - **Alarm Tab:** Language: Language 1, Text: <24H:MM:SS> <dd/mm/yy> Estop initiated Loading Belt 1
 - **Options:** ☒ Show in Alarm Objects, ☒ Require Confirmation, ☒ Save file to SD1
 - **Display:** ☒ Text Color: [Black], ☐ Blink; ☒ Back Color: [Red], ☒ Blink

Step 3: Screen Change Action Configuration
 - **Sequence List:** 01 - Alarm, 02 - Message Box, 03 - Screen Change, 04 - Sound
 - **Screen Change Tab:** ☐ Power Up Screen, ☐ Forward Screen, ☐ Previous Screen, ☒ Other Screen
 - **Other Screen:** Screen No: 1, Screen Name: Loading Belts

Step 4: Sound Action Configuration
 - **Sequence List:** 01 - Alarm, 02 - Message Box, 03 - Screen Change, 04 - Sound
 - **Sound Tab:** ☐ Beep, ☒ Sound: C:\NOTIFY_ADC001.WAV
 - **Repeat:** ☒ Loop, ☐ Times: 1

Example 2