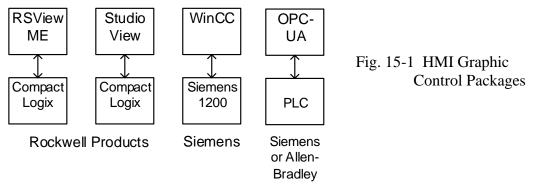
# Chapter 15 HUMAN MACHINE INTERFACE

## Introduction

Communications between processor and HMI (human machine interface) is an important subject as well as constructing an operator interface. The chapter includes procedures for attaching computers as HMI devices to the CompactLogix processor from A-B and the Siemens 1200 processor. Graphic control packages used are A-B's RSView ME and Siemens' WinCC. Other packages exist and were not excluded based on their capabilities. The ones used are among the more common and popular ones used today.

The following HMI panels communicating to PLC processors will be discussed in this chapter followed by a lab that can show advantages and disadvantages of each.



In each case, the emphasis is on getting a simple application operational and then expanding from that, remembering the analogy of the kite flying with a simple string over Niagara River. Later, the more difficult applications are discussed but only after a single button is programmed from the HMI and communicates successfully to the PLC.

## **Historical Panel Design**

The design of an operator panel requires much coordination with the programming of the PLC and the design of the machine being controlled. Before the computer-designed systems, there were individual component systems that were hard-wired to the control devices inside the panel.



Fig. 15-2 A Simple Control Panel with Push Buttons and Switches with Indicator Lights

Ch 15 Human Machine Interface



Fig. 15-3 This Printer is used for alarms for a process. Each alarm was recorded at the time of occurrence and printed as a single line of data to be analyzed by a process engineer or controls engineer.



Fig. 15-4 This panel shows many discrete devices as well as mimic panels showing process lines.

Meters show levels or flows of various devices.

Alarms are shown in grids of illuminated push buttons.



Fig. 15-5 Alarm panels were designed with discrete panels that lit or blinked with each alarm. Buttons were used to acknowledge each alarm point.



Fig. 15-6 Data was collected with recording devices similar to the above. Multiple points were individually recorded and studied.



Fig. 15-7 A handheld thermocouple readout device with paper recording output



Fig. 15-8 Several discrete controller devices for process control. Each device is capable of solving a single or multiple loops of process data executing a PID formula and controlling the output of the control loop.

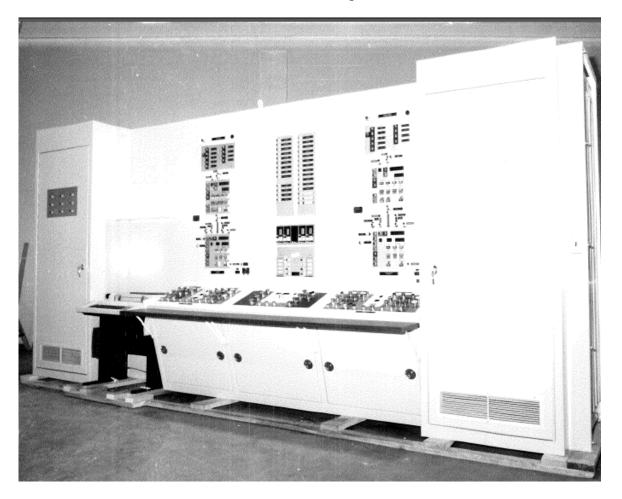


## Fig. 15-9

Whatever the appearance of the outside of the panel, the inside many times looks similar to the panel shown at left. It is too easy for the panel to look like this after a short time even though the original plan showed a neat design with well-organized wiring layout. This is not just a rare bad example.

## Fig. 15-10

The picture below is of a panel designed in 1980 for a chemical batching system. It should be seen as what 'was' and not as what today should be a good design. It cost approximately \$100K then and was a divider between the operator room and the instrument control room for the process.



## ISA-101 Human Machine Interfaces for Process Automation Systems

ISA-101 (officially ANSI/ISA-101.01-2015) is the HMI specification meant to describe proper development of an HMI panel. The specification is under development and should be referenced as it may have input for your HMI design.

## Hardware to Complement the Software

The following Siemens panels are available for selection to replace the various panels above. Software allows the design of these 'soft' panels to be adapted to represent all the designs shown above. Siemens, which provides a complete offering of hardware operator interface units to complement the software requirements.

And - Territory	States Transition	
		A B C 🔜 🗮 🛃 D 4

	KP8 PN	KP8F PN	KP32F PN		
Type of operation					
Function keys (programmable)	8	8	32		
Output type					
LED color modes	5 (green, red, yellow, blue, white)				
Typical service life					
Short-stroke keys (in number of switching cycles)	1,500,000				
Light-emitting diodes (ON period in %)	100%				
Interfaces					
Digital inputs / outputs1)	8	8	16		
Fail-safe inputs SIL 2 / SIL 3	-1-	2/1	4/2		
PROFINET	2	2	2		
Functionality					
Pushbutton and lamp test	•				
Degree of protection					
Front / rear	IP65 / IP20				

Fig. 15-11 Siemens HMI

	<b>***</b> ₩ 8	*** ® 8	**	**	**
	KTP400 Basic	KTP700 Basic DP KTP700 Basic	KTP900 Basic	KTP1200 Basic DP KTP1200 Basic	KP300 Basic mono PN KP400 Basic color PN
Operating mode	4" Touch + Key	7" Touch + Key	9" Touch + Key	12" Touch + Key	3.6" Key 4" Key
Display	Widescreen TFT, 65k	colors, LED backlighting			FSTN-LCD Black&White Widescreen TFT
Size (in inches)	4.3"	7"	9"	12.1"	3.6" 4.3"
Resolution (W x H in pixels)	480 x 272	800 x 480	800 x 480	1,280 x 800	240 x 80 480 x 272
MTBF <sup>6)</sup> backlighting (in h)	20,000	20,000	20,000	20,000	50,000
Front dimensions (in mm)	141 x 116	214 x 158	267 x 182	330 x 245	165 x 97 150 x 186
Operator controls	Touch screen and tactile keys	Touch screen and tactile keys	Touch screen and tactile keys	Touch screen and tactile keys	Tactile keys
Function keys (programmable) / system keys	4 / -	8/-	81-	10/-	10/• 8/•
Usable memory					
User memory	10 MB	10 MB	10 MB	10 MB	1 MB
Memory for options / recipes 4)	-/ 256 KB	– / 256 KB	– / 256 KB	– / 256 KB	- / 40 KB
Alarm buffer	•	•	•	•	•
Interfaces					
Serial / MPI / PROFIBUS DP / PROFINET (Ethernet)	-1-1-1•	• <sup>3)</sup> / • / • / - - / - / - / •	-1-1-1•	• <sup>3)</sup> / • / • / - -/-/-/ •	-1-1-1 •
USB host / USB device	1/-	1/-	1/-	1/-	-
Slot for CF / Multimedia/SD	-1-1-	-1-1-	-1-1-	-1-1-	-1-1-
Functionality (when configured with Win	nCC TIA Portal)				
Signaling system (number of messages / message classes)	1,000/32	1,000/32	1,000 / 32	1,000 / 32	200/32
Process pictures	100	100	100	100	50
Tags	800	800	800	800	250 500
Vector graphics	•	•	•	•	•
Bar charts / trend diagrams	• / f(t)	•/f(t)	• / f(t)	• / f(t)	• / f(t)
Faceplates	-	-	-	-	-
Recipes	50	50	50	50	5
Archiving / Visual Basic scripts	•1-	•1-	•1-	•1-	-1-
PG functions	-	-	-	-	-

Fig. 15-12 Siemens HMI

	SIMATIC HMI Comfort Panels						
**	** NOX **			*** ***			
KTP400 Comfort KP400 Comfort	TP700 Comfort KP700 Comfort	TP900 Comfort KP900 Comfort	TP1200 Comfort KP1200 Comfort	TP1500 Comfort KP1500 Comfort	TP1900 Comfort	TP2200 Comfort	
4" Touch + Key 4" Key	7" Touch 7" Key	9" Touch 9" Key	12" Touch 12" Key	15" Touch 15" Key	19" Touch	22" Touch	
		Widescreen TFT,	16 million colors, LED backlight	ing			
4.3"	7*	9"	12.1"	15.4"	18.5"	21.5"	
480 x 272	800 x 480	800 x 480	1,280 x 800	1,280 x 800	1,366 x 768	1,920 x 1,080	
80,000	80,000	80,000	80,000	80,000	50,000	30,000	
140 x 116 152 x 188	214 x 158 308 x 204	274 x 190 362 x 230	330 x 241 454 x 289	415 x 310 483 x 310	483 x 337	560 x 380	
Touch screen or tactile keys	Touch screen or tactile keys	Touch screen or tactile keys	Touch screen or tactile keys	Touch screen or tactile keys	Touch screen	Touch screen	
4 (with LED) / - 8 (with LED) / •	- / - 24 (with LED) / •	- / - 26 (with LED) / •	-/- 34 (with LED) / •	- / - 36 (with LED) / •	-1-	-1-	
4 MB	12 MB	12 MB	12 MB	24 MB	24 MB	24 MB	
4 MB / 512 KB	12 MB / 2 MB	12 MB / 2 MB	12 MB / 2 MB	24 MB / 4 MB	24 MB / 4 MB	24 MB / 4 MB	
•	•	•	•	•	•	•	

Fig. 15-13 Siemens HMI

• 3) / • / • / 1	• 3) / • / • / 2	• 3) / • / • / 2	• 3) / • / • / 2	• 3) / • / • / 3	• 3) / • / • / 3	• 3) / • / • / 3
1/1	2/1	2/1	2/1	2/1	2/1	2/1
-1•1•	-1•1•	-1•1•	-1•1•	-1•1•	-1•1•	-1•1•
2,000/32	4,000 / 32	4,000 / 32	4,000 / 32	6,000/32	6,000 / 32	6,000 / 32
500	500	500	500	750	750	750
1,024	2,048	2,048	2,048	4,096	4,096	4,096
•	•	•	•	•	•	•
• / f(t), f(x)	<ul> <li>/ f(t), f(x)</li> </ul>	<ul> <li>/ f(t), f(x)</li> </ul>	• / f(t), f(x)	• / f(t), f(x)	• / f(t), f(x)	• / f(t), f(x)
•	•	•	•	•	•	•
100	300	300	300	500	500	500
•/•	•/•	•/•	•/•	•/•	•/•	• / •
			STATUS / CON	TROL, diagnostics viewer		

	SIMATIC	HMI Mobile Pane	ls	
2 <sup>nd</sup> Generation	2 <sup>nd</sup> Generation	2 <sup>nd</sup> Generation		
KTP400F Mobile	KTP700 Mobile KTP700F Mobile	KTP900 Mobile KTP900F Mobile	Mobile Panel 277 IWLAN V2 Mobile Panel 277 (F) IWLAN V2	
4" Touch + Key	7" Touch + Key	9" Touch + Key	8" Touch + Key	Operating mode
Widescreen T	FT, 16 million colors, L	ED backlighting	TFT display (LCD), 64k colors	Display
4.3"	7"	9"	7.5"	Size (in inches)
480 x 272	800 x 480	800 x 480	640 x 480	Resolution (W x H in pixels)
50,000	50,000	50,000	50,000	MTBF <sup>6)</sup> back- lighting (in h)
194 x 166	248 x 172 248 x 195	307 x 201 307 x 224	Diameter 290	Front dimensions (in mm)
Touch screen and tactile keys	Operator controls			
4 (with LED) / -	8 (with LED) / -	10 (with LED) / -	18/-	Function keys (program- mable) / system keys
				Usable memory
4 MB	12 MB	12 MB	6 MB	User memory
4 MB / 512 KB	12 MB / 2 MB	12 MB / 2 MB	1,024 KB / 64 KB	Memory for options / recipes 4)
•	•	•	•	Alarm buffer
				Interfaces
-/-/1	-/-/1	-/-/1	- / - / - / (• via IWLAN)	Serial / MPI / PROFIBUS DP / PROFINET (Ethernet)
1/-	1/-	1/-	•1-	USB host / USB device
-1•1•	-1•1•	-1•1•	-1•1•	Slot for CF / Multimedia / SD
			Functionality (when configured	with WinCC TIA Portal)
2,000 / 32	4,000 / 32	4,000 / 32	4,000 / 32	Signaling system (number of messages / message classes)
500	500	500	500	Process pictures
1,024	2,048	2,048	2,048	Tags
•	•	•	•	Vector graphics
•/f(t), f(x)	• / f(t), f(x)	• / f(t), f(x)	• / • f(t)	Bar charts / trend diagrams
•	•	•	•	Faceplates
100	300	300	300	Recipes
•/•	•1•	•/•	•/•	Archiving / Visual Basic scripts
STATU	S / CONTROL, diagnosti	cs viewer	STATUS / CONTROL	PG functions

Fig. 15-14 Siemens HMI Ch 15 Human Machine Interface

	<b>Q</b> 1	<b>Ø</b> -1						
General features	Panel PC, 7" Touch	Panel PC, 9* Touch	Panel PC, 12" Touch	Panel PC, 15" Touch	Panel PC, 19" Touch			
Resolution in pixels (widescreen)	(800 x 480)	(800 x 480)	(1,280 x 800)	(1,280 x 800)	(1,366 x 768)			
Processor	Int	el Celeron N2807 (2C/2T, 1.58 (2.16)	GHz, 1 MB cache, VT-x); Intel Celeron	N2930 (4C/4T, 1.83 (2.16) GHz, 2 MB cac	he, VT-x)			
Main memory		2 GB, 4 GB or 8 GB; 512 KB NVRAM optional						
Free expansion slots		-						
Operating systems (preinstalled and activated)		Windows Embedded Standard 7 (EIP), 32-bit/64-bit; Windows 7 Ultimate, MUI <sup>11</sup> , 32-bit/64-bit						
Packages / bundles		Packag	es with WinCC RT Advanced, WinCC V7	and WinAC RTX (F)				
Power supply / temporary voltage Interruption		24 V DC; 20.4 28.8 V; isolated / max. 10 ms (according to NAMUR); OniOff switch						
MTBF backlighting		up to 80,000 h <sup>n</sup> ; din	mable from 0 to 100%		up to 50,000 h <sup>n</sup>			
Drives								
Mass storage		CFast up to 16 GB (a	ccessible from outside); SSD 80 / 160 GB;	HDD 320 GB (IPC227E only)				
Optical drives			-					
Interfaces								
Fieldbus			PROFINET RT over Ethernet					
Ethernet			2 x 10/ 100/ 1000 Mbps (RI45);	teaming				
JSB	Rear: 1 x USB 3	.0, 2 x USB 2.0	Rear: 1 x USB 3.0, 3 x USB 2.0	Rear: 1 x USB 3.0, 3 x	x USB 2.0; front: USB 2.0			
Serial / parallel			1 x RS 232/RS 485/RS 422 can be selected	in BIOS				
Graphics interface			1 x DisplayPort					

		SIMATIC IPC47	7D	
- <b>Q</b> . AI			75	
Panel PC, 12" Touch	Panel PC, 15* Touch or Multitouch	Panel PC, 15" Touch/Key	Panel PC, 19" Touch or Multitouch	Panel PC, 22" Touch or Multitouch
(1,280 x 800)	(1,280 x 800)	(1,280 x 800)	(1,366 x 768)	(1,920 x 1,080)
	Intel Core i7-3517UE (2C/4T, 1.7 (2.8) GHz, 4 M	MB cache, VT-x/d, AMT); Intel Core i3-3217UE (	2C/4T, 1.6 GHz, 3 MB cache, VT-x); Intel Celeron 827E (1.4	GHz, 1.5 MB cache, VT-x)
		1 GB, 2 GB, 4 GB or 8 GB; 512 KB	NVRAM optional	
2		1 x PC	le x 4 (optional); max. 5 W	
		VxWorks Support Package, su	ited for Linux	
	Bundles with WinC	C RT Advanced and WinAC RTX (F); packages:	WinCC RT Professional, WinCC V7, WinAC RTX (F)	
24 V DC; 19.2 28.8 V; isolated / max. 20 ms (according to NAMUR); On/Off switch		24 V DC; 19.2 28.8 V; isolated / max. 20 ms	(according to NAMUR); or 100-240 V AC, 50/60 Hz; On/	Off switch
	up to 80,000 h <sup>19</sup> ; dimmable from 0 to 100	%	up to 50,000 h	up to 30,000 h
CFast up to 16 GB (accessible from outside); SSD 80 / 160 GB		CFast up to 16 GB (accessi	ole from outside); SSD 80 / 160 GB; HDD 320 GB	
-			DVD-RW (optional)	
	PROFINET RT over Ethernet; PR	OFIBUS DP/MPI: 1 x 12 Mbps (isolated) option	al; PROFINET IRT: 1 x 10/100 Mbps (3-port-switch, option	nal)
	2 x 10/1	100/1000 Mbps (RI45); teaming; 1 x 10/100/	1000 Mbps with PROFINET IRT variant	
Rear: 4 x USB 3.0		Rear: 4 x USB 3.0; fr	ont: 1 x USB 2.0 (with Touch version only)	
		COM1: RS 232; COM2: RS 23	(optional)	
		1 x DVI-I; 1 x Display		

SIMATIC IPC477D

#### SIMATIC IPC677D



Panel PC, 15", 19" or 22" Touch or Multitouch	General features
15" Touch: 1,366 x 768; 15" Multitouch: 1,280 x 800; 19": 1,366 x 768; 22": 1,920 x 1,080	Resolution in pixels (widescreen)
Intel Xeon E3-1268L v3 (4C/8T; 2.3 (3.3) GHz; 8 MB cache; VT-d; AMT 9.0); core i3-4330TE (2C/4T; 2.4 GHz; 4 MB cache; VT-x); Celeron G1820TE (2C/2T; 2.2 GHz; 2 MB cache)	Processor
From 2 GB DDR3-1600 SDRAM; 2 x DIMM; configurable up to 16 GB; ECC optional; non-volatile memory: NVRAM 2 MB optional	Main memory
2 x PCI (240 mm) or 1 x PCIe x 16 (185 mm), 1 x PCI (185 mm) or 1 x PCIe x 16 (185 mm), 1 x PCIe x 4 (185 mm)	Free expansion slots
Windows 7 Ultimate (32/64-bit) MUI <sup>11</sup> ; Windows Embedded Standard 7 P (32-bit); released for S7-1500 Software Controller, suited for Linux	Operating systems (preinstalled and activated)
Packages with WinCC V7; WinCC RT Advanced; WinCC RT Professional and WinAC RTX (F)	Packages / bundles
AC: 100-240 V; 50-60 Hz / max. 20 ms (according to NAMUR); 24 V DC: 20.4 28.8 V	Power supply / temporary voltage interruption
up to 50,000 h	MTBF backlighting

Fig. 15-15 Siemens HMI Ch 15 Human Machine Interface

	SIMATIC Indus	trial Monitors and Thin Clie	nts
SIMATI	C Industrial Thin Client	SIMATIC Inc	dustrial Flat Panel
General features	12", 15", 19" or 22" Touch	15" Touch or Multitouch, 15" Key	19" and 22" Touch or Multitouch
Resolution in pixels (widescreen)	12" (1,280 x 800) 15" (1,280 x 800) 19" (1,366 x 768) 22" (1,920 x 1,080)	15" Multitouch: (1,366 x 768) 15" Touch: (1,280 x 800) 15" Key: (1,280 x 800)	19": (1,366 x 768) 22": (1,920 x 1,080)
Max. distance to PC	Unlimited over Ethernet	Standard: 5 m; extended: 30 m	Standard: 5 m; extended: 30 m
Processor	Intel Celeron (1.2 GHz)		-
Operating system (preinstalled and activated) / supported protocols	Closed Linux / VNC; SINUMERIK; WinCC-OA; web browser; JAVA; CITRIX Client	-	-
Power supply / max. power consumption	12" approx. 28 W 15" approx. 36 W 19" approx. 32 W 22" approx. 53 W	15": 24 V DC; 19.2 28.8 V, approx. 40 W	24 V DC; 19.2 28.8 V, approx. 40 W
MTBF background lighting	up to 50,000 h <sup>m</sup> ; dimmable from 0 to 100%	up to 50,000 h <sup>77</sup> ; dimmable from 0 to 100%	up to 50,000 h <sup>23</sup> ; dimmable from 0 to 100%

Fig. 15-16 Siemens HMI

The hardware above shows the variable nature of hardware panels available for use. The cell phone is also available with an app installed that communicates with the system software. Each device gives a level of access to the PLC program and give visual feedback to the engineer or plant personnel.

## The Software for Development

We begin the software development of the three programs. Two programs exist in the A-B platform and one is available from Siemens. We will also discuss an OPC-UA application for a total of four different programming platforms.

## **Common Tags**

The most obvious advantage of using any new PLC platform is the universal accessibility of data tags. Tags created in any tool for any device are automatically and immediately accessible to other devices. If, for example, a user creates a new tag in the PLC to measure temperature, that tag is automatically created in the operator panel at the same time. This saves valuable engineering time compared to earlier methods that require the tag to be created in each device.

The suspension bridge at Niagara Falls was started by flying a kite with a string attached across the Niagara River. When wind conditions were favorable, the kite was flown across the river. Then a string was attached to the thread and a bridge was the eventual result. Likewise, programs in this chapter can be started with small threads and then expanded. It is best to get a simple device such as a button programmed and fully working and then adding the rest of the project after the button has been proved to thoroughly work in all modes

## Allen-Bradley's New View Designer

Allen-Bradley recently released a new graphic HMI Development Environment. It is found on Studio 5000 as 'View'.



We began the View Designer with a project named 't7' tied to a plc program in a processor named 'r4'.

Yiew Designer - t7 [10.4" PanelVie         File       Edit       View       Project       Comm         Image: Ima	nunications Arrange Tools Window Help	_ = ×
Project Explorer 🔹 म 🗙	Screen_001 +> ×	▼ Properties ▼ ₽ ×
🖬 t7 🔶	- 50% + 🖸	- Name: Screen_001
Mavigation Menu		Type: Screen
Screen_001		A A A A A A A A A A A A A A A A A A A
🔁 AlarmSummary		Properties Animations Events
🔁 AlarmManager 📃		* Add Animation •
👼 Settings		
User-Defined Screens	↓         ↓	
Screen_001		
Toolbox - I X		
▶ / □ ○ ♀ _		
Q▼ Search ▼		
Add-On Graphics		
Alarms		
b ButtonsAndSwitches		
Flags		
b GeneralEquipment		
Graph		
Security		Graphic Explorer 🛛 🔻 🕂 🗙
b SignsAndSymbols		Screen_001
System 💌	Rectary	gula <mark>r</mark> Snip
CommonControls		
123 Numeric Display		v .
1231 Numeric Input	< > Design / Property Definition	-
· · · · · · · · · · · · · · · · · · ·	Design / Property Definition	*

Ch 15 Human Machine Interface

The plc application is already running and the application is selected under Project properties. We select a 10.4 " PanelView 5500 panel although we are never going to download to this HMI panel. We plan to only use it in the emulate mode.

P	roject Pr	roperties			?	×
		<b>Application</b> Configure target HMI Device settings	Target HN Location: 5	II Device Select path to HMI device	~	
	Eo III I	<b>References</b> Create and configure controller references	Туре:	10.4" PanelView 5500 2715-B10Cx		

We then select the reference tab and tie the HMI to the processor 'r4' at IP address 192.168.0.4.

J	roperties		?	×
	<b>. .</b>	Controller References:		
	Application Configure target HMI Device settings	Controller Reference Name: r4 Logix Project File: C:\Users\wevans\Documents\Stud		×
	References Create and configure controller references	HMI to Controller Path:       192.168.0.4       Slot:       Direct          Emulator to Controller Path:       AB_ETHIP-3\192.168.0.4	•••	

We then start with a button but use a rectangle instead of the pre-designed buttons.

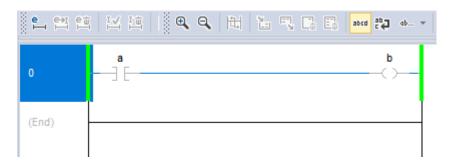
_		
- 1	() Unactived ← → ● ≗ Losg on	
	6—B	

Right click on rectangle. Choose Set tag to 1 on press, 0 on release. Then below is a box for the tag. If attached properly above to the processor, the available bit may be seen in the Tag window. Choose the appropriate variable.

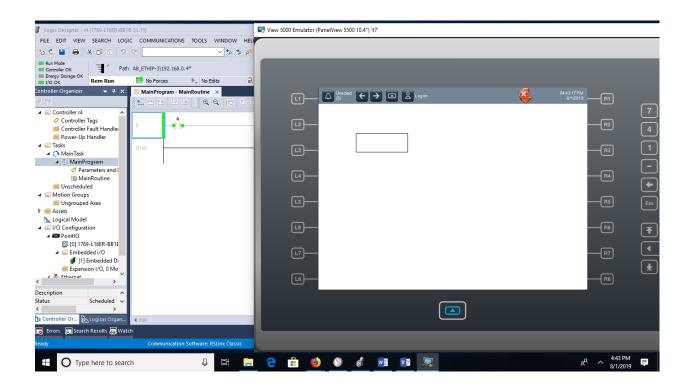
☆ Button Behavior ×
Set a tag to 1 on press, 0 🔻
Key: Touch Only Requires Focus Always Trigger Release
Tag:
Minimum Hold Time:

Ch 15 Human Machine Interface

The tag we tied to resides in the PLC and is ready to be tested. If we were to toggle the button (rectangle) on and off, we should see the contact respond accordingly with the emulator running.



Under View 5000's Project tab, run the emulator program. With the emulator running, by pushing the rectangle with the mouse button, the bit 'a' should turn on. It does. This is a good test that the program is communicating properly with the HMI and gives you a positive feedback that you are ready to proceed with other logic development. The button works!



In the state table for the rectangle (button), change the number of states to 2 and tie the states to 'b', the coil in the example. When the button is pushed, the color changes from green to orange. The button now reflects the status of the coil 'b'.

🔄 Order 🝷 🥼 Flip 🔹	- 50%	+	÷	Nan Type	e: Rectangle				
				:r4	StateTable_001 ression: \MainProgram.b			×  5	•
					Expression Value Default 0 1	State Name d Default State0 State1 State	FillColor #ffffff #00a651 #ff5a00		

We now have a button and indicator light similar to the ones used in the hard-wired system previously used. We have a button and illuminated indicator that may be used together or separately.

## The RSView ME Stand-Alone Application

RSView Studio is the older A-B product. It resides on the PC's in NE 2350 but not in NE 2390. To open RSView Studio and gain access to RSView ME, open the program and click on the New tab. For Application name, the following example program used test.

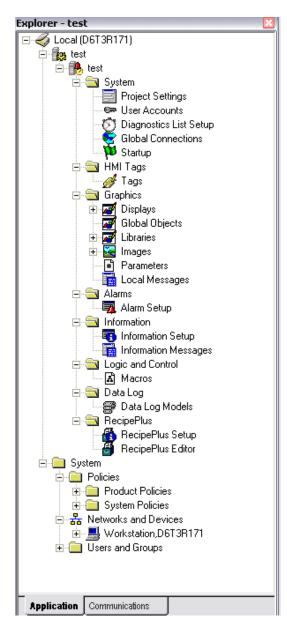
RSView Studio - Machine Edition		
File View Application Tools Window Help		
🗹 🖬 🚭   D 🖆 🥓 🎘		
Explorer 🛛		
	New/Open Machine Edition Application	
	New Existing	
	Application name: test	
	Description:	
	Language: English (United States), en-US	
	Create Import Cancel	
	RSView Studio	

The A-B product uses RSLinx Enterprise. This means that RSLinx Enterprise has the ability to allow this software package to browse directly for the tag database and link existing tags and not requiring a tag created in the HMI to complement the tag in the PLC.

When starting a new application, select Objects from the main menu. Notice the types of objects Ch 15 Human Machine Interface 14 selectable. Under each selection are sub-menu selections. In the case of Push Buttons, the four sub-menu selections are:

Momentary Maintained Latched Multistate etc

The pushbutton is a good example of a type of object that can be used for a number of different operations. For example, momentary is the most used button with an exact simulation of a real button in which the operator pushes the button and a '1' appears in the data address of the device. When the button is released, the button returns to a '0'. As simple as this is, it is a very profound device in that the device continues to write a '1' to the data tag until the device de-activates and the device writes a '0'. Other button types have various other characteristics and these characteristics allow the programmer flexibility in programming of the various functions surrounding the logic of the pushbutton. More will be discussed later regarding some of the button types.



RSView Machine Edition (ME) is designed for the machine-level HMI and supports operator interface solutions for the monitoring and controlling of individual machines or for small processes.

The system tree at left shows the graphical application and is organized by area. These include:

System HMI Tags Graphics Alarms Information Logic and Control Data Log RecipePlus

For the simple assignments of this text, the user can simply add a display and create the appropriate graphics on the form. To test or run the form requires a link to a PLC. The procedure for linking to a PLC is included next. The tags created for the PLC are used in the graphic and any button or indicator will reference the PLC tag.

Ch 15 Human Machine Interface

Remember that input and output bits used for a HMI are tags with internal addresses, not hardwired inputs or outputs. For A-B, the device should have a tag referring to an internal location. For Siemens, the address should have an M prefix.

-			1
Rockwell Software			
PactoryTalk Administration Consol	e	Pictures	
BOOTP-DHCP Server			
FactoryTalk Activation		Music	
퉬 FactoryTalk Gateway			
FactoryTalk Tools		Computer	
Diagnostics Viewer			
FactoryTalk Alarms and Events Syste		Control Panel	
🏀 餐 FactoryTalk Directory Configura	ation _	Devices and Printers	
😵 FactoryTalk Help	Factory	Talk Directory Configuration	Wizard
🔁 FactoryTalk Security System Co	Configu	ure the FactoryTalk Network	Directory or FactoryTalk Local Directory
Import RSSecurity Configuration	on this	computer.	
🖙 Log On to FactoryTalk		Help and Support	
📲 Rockwell Software Data Client			
😽 RSSecurity Emulator Install			
🍓 Specify FactoryTalk Directory Lo	ocati		
Windows Firewall Configuration	n Util 👻		

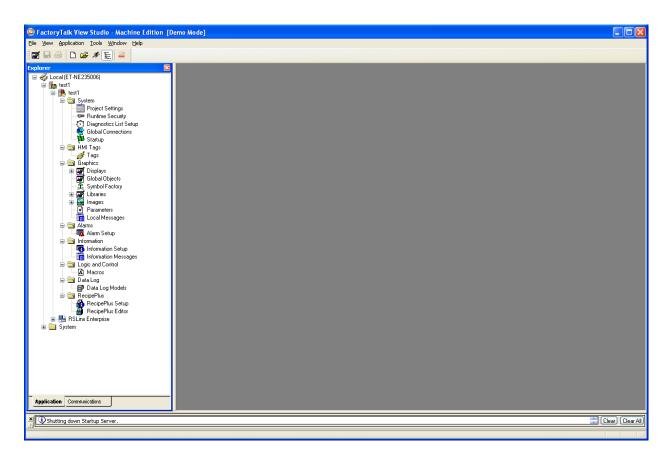
Then, configure the local directory per administration sign-in configuration of your pc.

Reconfigure FactoryTalk Local Directory						
Reconfiguring the FactoryTalkLocal Directory allows you to reset a disabled administratoraccount and upgrade policies.						
To continue, enter a Windows Administrator account user name and password below.						
Administrator User Name:	plcadmin					
Password:	****					

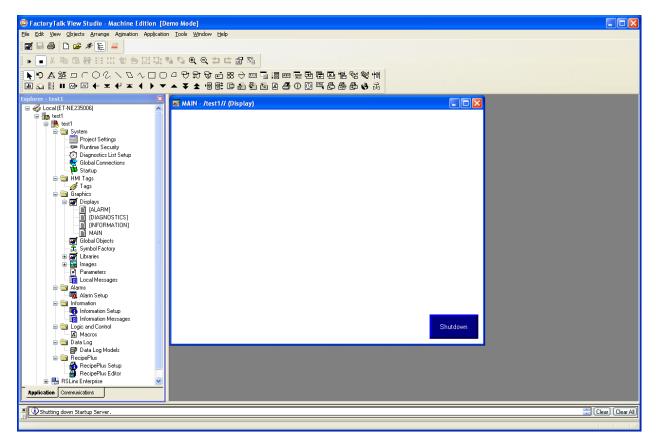
After this configeration is complete, you may proceed with your HMI application and connect the HMI screens to a PLC via RSView Enterprise as described above.

Creation of tags is accomplished either in the program tag or controller tag areas. Either may be accessed by FactoryTalk. The type of tag must be considered since numeric tags must be scaled and configured for proper display. Boolean tags must also be configured properly.

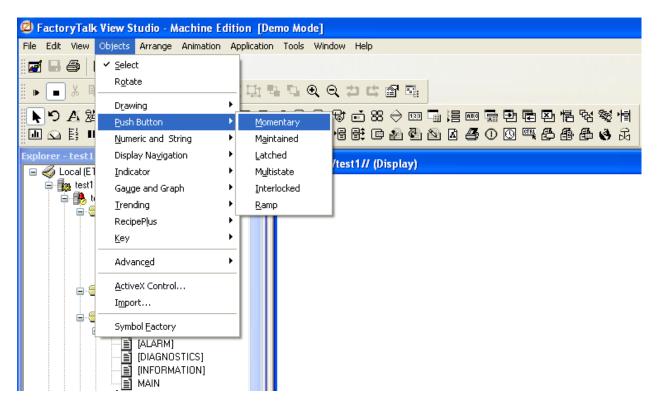
The project is begun with the following screen present. The project name is 'test1'.



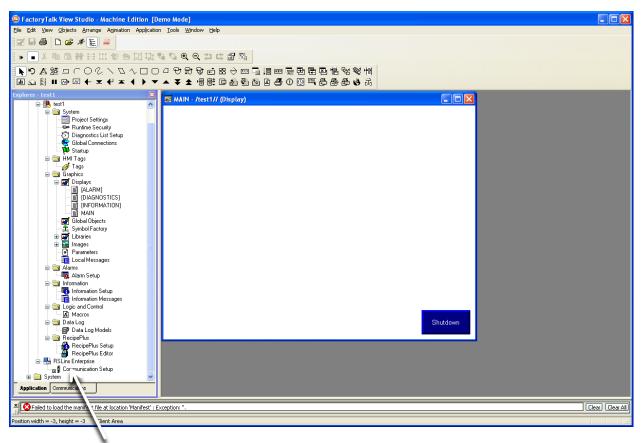
First Screen with RSView Studio



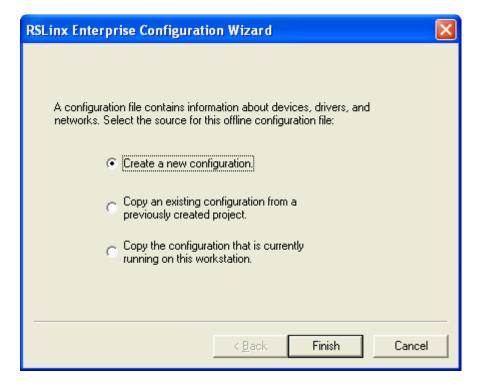
Selecting the Main Display



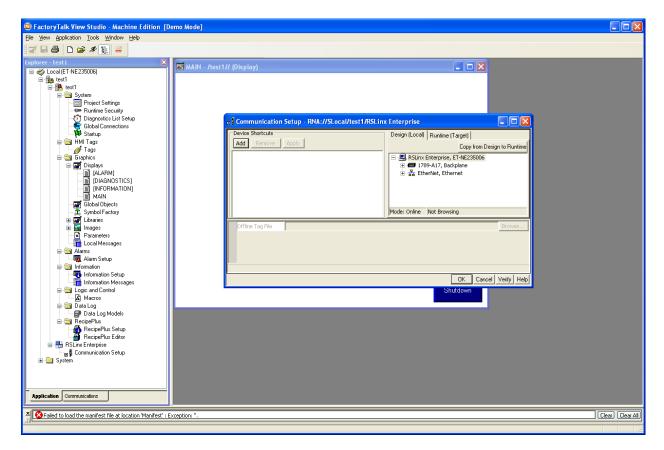
Building a Momentary Push Button



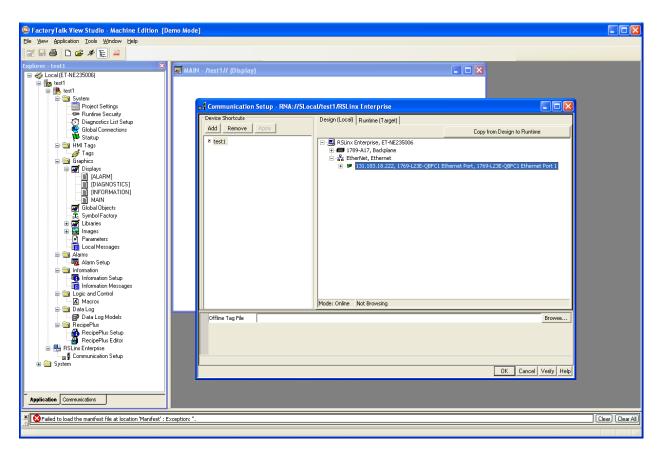
Establishing Communication with RSLinx Enterprise



After Selecting Communication Setup under RSLinx Enterprise



Click 'Add' for Device Shortcut, Enter Name 'test1'



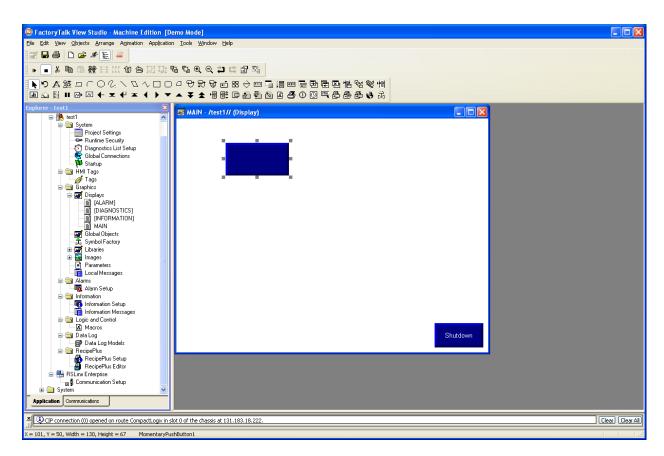
Highlight the Ethernet Device – PLC When Apply IS HIGHLIGHTED - CLICK IT

RSLinx Enterprise					
You've made the following changes to the shortcut 'test1':					
Design (Local) path edited - Old:					
<ul> <li>New: CompactLogix System 4.Motor_1_Starter</li> </ul>					
Press Yes to apply changes. Press No to discard changes.					
<u>Yes</u> <u>N</u> o					

The Local Path is Determined for the Application

FactoryTalk View Studio - Machine Edition [Demo Mod	e]			
Eile View Application Tools Window Help				
🗹 🖬 🖨 🗅 🚅 🚿 🔃 🚔				
Explorer - test1	N - /test1// (Display)			
🔄 🚽 🌍 Local (ET-NE235006)	n - Mest M (Display)			
🛱 📴 test1				
🗐 🚍 System				
Project Settings				
- 🚫 Diagnostics List Setup	Communication Setup - RNA://SLo	cal/test1/RSLinx Enterprise		
Global Connections	Device Shortcuts	Design (Local) Runtime (Target)		
😑 🚖 HMI Tags	Add Remove Apply		Copy from Design to Runtime	
∰ Tags ⊡∰ Graphics	test1	E RSLinx Enterprise, ET-NE235006		
😑 🌌 Displays		<ul> <li>IT89-A17, Backplane</li> <li>Backplane</li> <li>Backplane</li> </ul>		
(ALARM) [DIAGNOSTICS]		🖻 - 📂 131.183.18.222, 1769-L23E-QBFC	1 Ethernet Port, 1769-L23E-QBFC1 Ethernet Port 1	
INFORMATION]		🖻 🏢 CompactLogix System, Compac 📄 👔 0, 1769-L23E-QBFC1, Moto	tLogix System 4	
MAIN		🗄 💑 DF1P2P, DF1 Point-to-F	Point 2	
Global Objects		🗌 3, Local Adapter, VA1769//	A	
🕀 🐷 Libraries				
Images     Parameters				
Local Messages				
alarms				
🖃 🔄 Information				
Information Setup     Information Messages				
😑 🔄 Logic and Control				
🔲 📓 Macros 🖃 🔄 Data Log				
Bata Log Models		Mode: Online Not Browsing		
🖮 🔄 RecipePlus	Offline Tag File	199 <u>7</u>	Browse	
RecipePlus Setup	or more regime			
😑 🖶 RSLinx Enterprise				
🕞 🖉 Communication Setup				
			OK Cancel Verify Help	
Application Communications			Save changes and notify RSLinx	Enterprise of change
			/^	
Failed to load the manifest file at location 'Manifest' : Exception: '	•			Clear Clear All

The Local Path is set with 'OK'



Back at the Ranch (I mean Button)

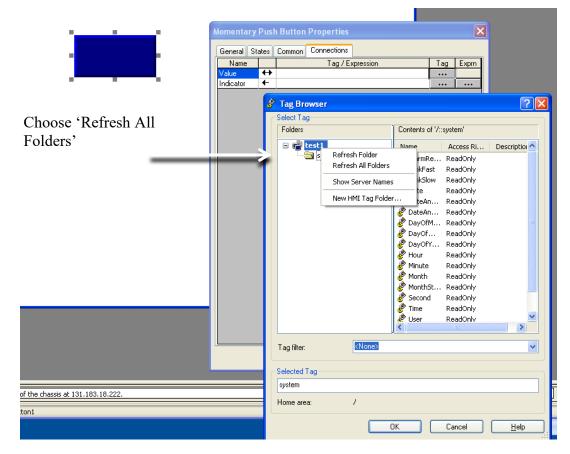
Momentary Push Butto	n Properties		×
General States Common Appearance Border style: Raised V Back style: Solid V Shape: Rectangle V	·	<ul> <li>✓ Border Uses Back Color</li> <li>Highlight color</li> </ul>	
Button settings Button action Normally open Hold time: 250 msec	O Normally closed	<b>O</b> Value	
Touch margins Horizontal margin: 0 Other V Audio	Vertical margin:		
	OK Cancel	Apply Help	

Working with the Button – Tab 1 - General

lomentary	Pusi	1 Button Properties		Z
General S	tates	Common Connections		
Name		Tag / Expression	Tag	Exprn
Value	↔		•••	
Indicator	+		+++	•••
		OK Cancel Apply		Help

Establishing the PLC Connection to the Button

🗳 Tag Browser				? 🛛			
Select Tag							
Folders	Folders			Contents of '/'			
Folders <b>test1</b> ∴ - → system		Name	Access Ri	Description			
Tag filter:	<none></none>			~			
Selected Tag							
Home area:	7						
		ж	Cancel	<u>H</u> elp			



Ch 15 Human Machine Interface

🔗 Tag Browser			? 🛛
Select Tag			
Folders	Contents of V::	test1/Online/Pro	ogram:MainProgr
	Motor_1 Motor_1 motorstart	Access Ri ReadWrite ReadWrite ReadWrite ReadWrite ReadWrite	Description
Tag filter:			~
Selected Tag			
::[test1]Program:MainProgram			
Home area: /			
	ж	Cancel	<u>H</u> elp

PLC Tags are Shown in the Online Tags

Momentary F	Push	Button Properties		
General Sta	tes C	Common Connections		
Name		Tag / Expression	Tag	Exprn
Value	↔ {	:::[test1]Program:MainProgram.stop}	•••	
Indicator	←		+++	•••
				_
		OK Cancel Apply		Help

The Tag shows the Tag/Expression with the PLC Tag

	Button Properties	Color and Captions Modified under States Tab
Select state: State0 State1 Error	Bink Pattern color	
	Caption	
	Insert Variable         Font:       Size:         Arial       Ill       IB       I       II         Caption color       Alignment       Caption back style       Caption back style         Caption back color       O       Transparent       Transparent	
Cgpy	Word wrap  Mage settings  Image: Image: Image: Image back style Image color Image back color Image blink OOO Image blink OO Image blink O Image blink Image blink O Image blink	
	Common Connections	The Common Tab Controls Size, Position and Name of the Button
Height: 67 Other	Width:         Top:         Left:           130         50         101	
Name: MomentaryPr	ushButton1	

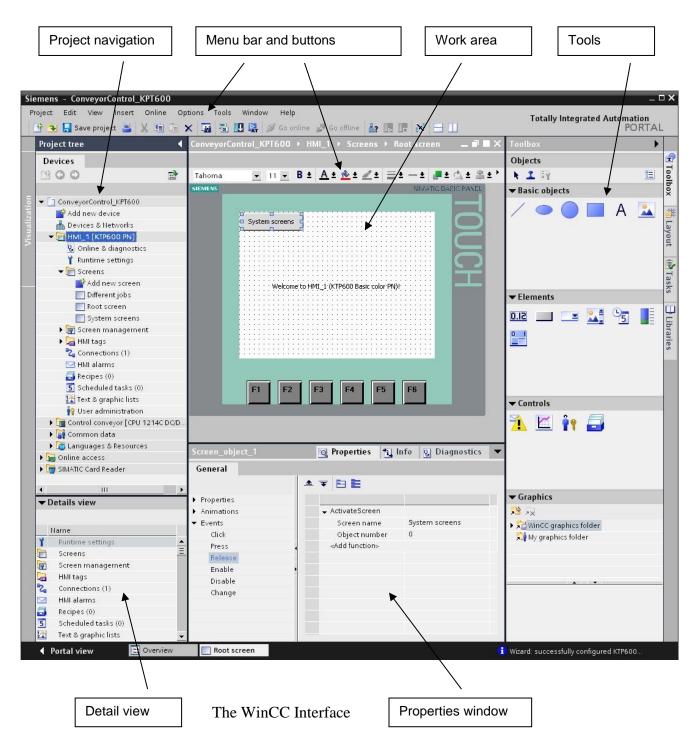
It is time to try the button with the connection to the PLC. Run the display by choosing the triangle in the command line. Test the screen with the button in the off and on position. View the bit in the PLC online program.



Use the triangle for testing single screens. To run all screens together in local mode, run the little man. There may be problems with this operation as it invokes a program called 'KEP' that may interfere with other applications. Be careful when running the little man.

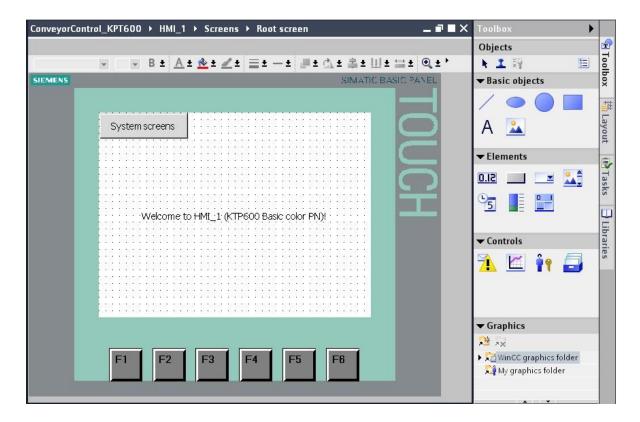
While building a screen or series of screens requires more instruction, we leave the A-B software to discuss Siemens' HMI interface.

## Siemens Win CC WinCC flexible Interface



#### Work Area

In the work area we edit the objects of the project. All elements of WinCC flexible are arranged around the work area. In the work area, we edit the project data either in the form of tables (for example, variables), or graphically (for example, a process display). The upper part of the work area contains a symbol bar. Here, the font, the font color or functions such as Rotate, Align, etc. can be selected.



The Root Screen

## Tools

The tool window provides a selection of objects that you can insert in your screens; for example, graphic objects and operating elements. In addition, the tool window contains libraries with assembled library objects and collections of picture blocks. The objects are dragged and dropped into the work area.

#### **Properties Window**

The properties of objects are edited in the properties window; for example, the color of screen objects. The properties window is available only in certain editors. The properties of the selected object are displayed in the properties window, arranged according to categories. Value changes become effective as soon as an entry field is exited. If you are entering an invalid value, it is color-enhanced.

By using QuickInfo, information is provided about the valid value range, for example. In the properties window, animations and events of the selected object are configured also; here, for example, a screen change when releasing the button.

Screen_object_1	9	Properties	🚹 Info	<b>9</b> Diagnostics	
General					
	▲ ∓ 🖹 🖹				
Properties					
<ul> <li>Animations</li> </ul>	🚽 ActivateScreen				
✓ Events	Screen name	Sy	stem screen	S	
Click	Object number	0			
Press	<add function=""></add>				
Release					
Enable					
Disable					
Change					

Dealing with Object Properties

#### **Details View**

In the Details view, additional details about the object highlighted in project navigation are displayed.

▼ D	<ul> <li>Details view</li> </ul>			
1	Name			
Y	Runtime settings			
	Screens			
P	Screen management			
	HMI tags			
2	Connections (1)			
	HMI alarms			
3	Recipes (0)			
5	Scheduled tasks (0)			
1	Text & graphic lists			
19	User administration			

Details of the Object

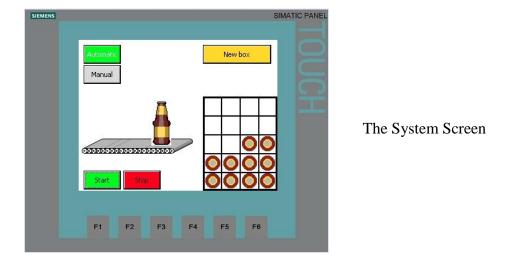
#### **Operating Screens and Connections**

A screen can consist of static and dynamic components. Static components, such as text and graphs, are not updated by the control system. Dynamic components are connected to the control system and visualize current values from the control system's memory. Visualization can be in the form of alphanumerical displays, curves and bars. Inputs at the operator panel that are written to the memory of the control system are also dynamic components. They are interfaced with the control system by means of variables. Initially, we are only generating a screen for our conveyor control.

#### **Root Screen or Start Screen**

This screen was set up automatically and defined as start screen. Here, the entire plant is represented. Buttons can be used to do the following: switching the operating mode between

automatic and manual; starting and stopping the conveyor motor, and exchanging the box. The movement of the bottle on the conveyor belt and the fill level of the box are represented graphically.



Using button F6, we are jumping to the system screen:

## **Connections to S7 Control Systems**

In the case of operator objects and display objects that access the process values of a control system, a connection to the control system has to be configured first. Here we specify how the panel communicates with the control system, and with which interface.

In Project navigation, click on Connections. Because of the settings in the hardware configuration, all parameters are already set.

Project tree	Conve	eyorControl_KPT60	0 → HMI_1 → Connec	ctions (1)			_ ∎ ■ ×	
Devices								
BOO B	the Co	onnections to S7 PLCs i	n Devices & Networks					
	Conn	ections						
- ConveyorControl_KPT600		Name	Communication driver	Station	Partner	Node	Online	
Add new device	ល័ង	HMI-Connection_1	SIMATIC S7 1200	SIMATIC 1200-Statio	Control Conveyor	CPU 1214C DC/DC/DC,		
Devices & Networks								
★ ☐ HMI_1 [KTP600 PN]								
😼 Online & diagnostics								
🍸 Runtime settings								<b>F</b> ( 11' 1 '
🕶 🛅 Screens	4		<u> </u>				•	Establishing
Add new screen	Para	ameter Area po	inters					the PLC
basic_screen	Tara	meter Area pe	inters					
further_tasks	L/T	FP600 Basic color Pl				Stati		Connection
system_screens	KI	POUD Basic color Pi	N			Stati	on	
🕨 🗑 Screen management		Interface	:			144 ( )		
🕨 🔚 HMI tags		PROFINE	T (X1) 👻			<u>di</u>		
Connections (1)								
🖂 HMI alarms								
📑 Recipes (0)	HM	MI device			PLC			
5 Scheduled tasks (0)		Address: 192	168 0 2		,	Address: 192 168 0	1	
🔛 Text & graphic lists					Evenance	ion slot: 1		
👔 User administration			ddress can only be config	ured on the	Lyans			
Control Conveyor [CPU 121.		HMI	device.			Rack: 0		
🕨 🙀 Common data		Access point: S70	ILINE					
Languages & Resources								
🕨 🔚 Online access								
🕨 🥅 SIMATIC Card Reader								
<u>∢                                     </u>								

The IP address has to be assigned to the panel also. By means of Accessible devices, read out the panel's MAC address, or read it on the back of the panel.

cccessible devices	Accessible devices in		to show acces:	sible devices for: 🏾 🌉 I	× ntel(R) PRO/100 VE Networl 💌	
	Device	Device type	Туре	Address	MAC address	
Flash LED	Control Conveyor	CPU 1214C DCID SIMATICHINI	TCP/IP TCP/IP	19216801 19216802	00-1C-06-00-6D-42 00-1C-06-01-9F-39	Assign IP Add
					<u>R</u> efresh	
				Show in project t	Close	

ng the ess

## **Assigning the IP Address**

After inputting the MAC address, the IP address can be assigned under Online & diagnostics. The panel has to be in the Transfer Mode in this case. Change the IP address to 192.168.0.5 for all applications.

Project tree 📢	ConveyorControl_KPT600	► KTP600 PN ► WinCC RT HMI		_∎×
Devices				
13 O O 12	Online access Functions	Assign IP address		_
- ConveyorControl_KPT600	Assign IP address			
Add new device		MAC address:	00 - 1C - 06 - 01 - 9F - 39	
📥 Devices & Networks		IP address:	192.168.0.2	
▼ ☐ HMI_1 [KTP600 PN]			255 . 255 . 255 . 0	
😵 Online & diagnostics				
🍟 Runtime settings		Use router:	192 168 0 1	
🗢 🛅 Screens	-			
📑 Add new screen	-		Assign IP address	
basic_screen				
🗾 further_tasks				
system_screens		- 1		
🕨 🕅 Screen management				
🕨 🔙 HMI tags		•		
🎝 Connections (1)				
🖂 HMI alarms	-			
🛃 Recipes (0)				
5 Scheduled tasks (0)				
🔛 Text & graphic lists				
📲 🛊 💡 User administration				
Control Conveyor [CPU 121				
🕨 🙀 Common data				
Languages & Resources				
🕨 🔚 Online access				
🕨 🣴 SIMATIC Card Reader				
▲ III				

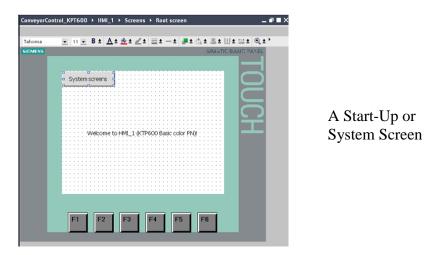
#### Note

The IP address can also be checked or entered on the panel in the system control under Control Panel at Profinet.

## **Configuring the Root Screen**

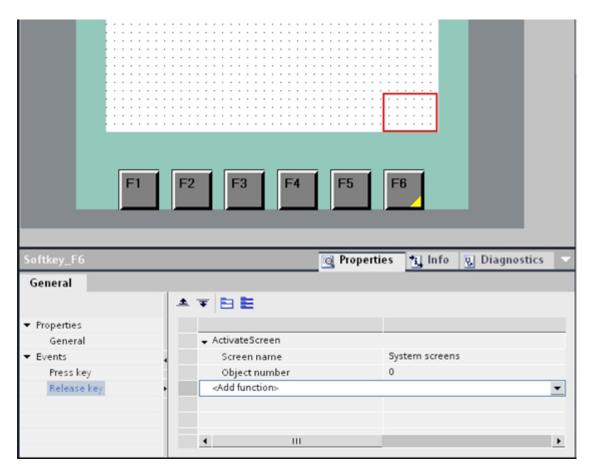
Clicking on the button "System screens" displays the system screen. We want to transfer the function of the button System screens to the function key F6.

Select System screens and in the Properties window below copy the function Activate screen at Events Release.



#### **Function Key F6**

Select function key F6 and in the Properties window below, insert the function Activate screen at Events Release key. Then, delete the text field in the center, and delete or remove the button System screens.



Defining a Function Key

The yellow corner on the F6 key indicates that the key is configured.

## Configuring the Buttons Automatic and Manual

Drag a button into the work area of the root screen.

SIEMENS SIMATIC BASIC PANEL	▼ Basic objects
Text x/y: 8,8	Α 🔽
w/h: 60.30	▼ Elements
	🛴 💶 💶 📶
	© <u>5</u> ∎ ∎
	✓ Controls
	<u>71</u> 🖾 🛉 🔒
	<b>▼</b> Graphics
	🔁 📈
F1 F2 F3 F4 F5 F6	▶ 2 WinCC graphics folder 2 My graphics folder

Configuring a Button

At Label, enter Automatic.

**Caution!** Don't press the input key; otherwise, a second line is generated.

Button_1		💽 Prope	rties 📆 In	ifo 😨 Diagnosti	cs 🗸 🗸
General					
	General				
<ul> <li>Properties</li> <li>General</li> </ul>	Mode	Label			
Appearance	• Text	• Text	🕘 Text li	st	
Design	Graphic	Off	Automatic	*	
Layout	Graphics and text			*	
Text format	•		os: 70	1000	
Miscellaneous	🔘 Invisible	On	Text	*	
Security				*	
Animations	Hotkey				
Events	Hotkey: None 💌	T.			

Under Layout, enter position and size.

	Layout							
Properties	Desition	0 -!	3			-		
General	Position	& SIZ	e				Fit to size	
Appearance	X:	8	-	₩	60	-	Fit object to contents	
Design	¥:	8	\$	Ī	30	-		
Layout		a.	2.22310		30	CONTRACTOR OF		
Text format								

Under Events Press select the function SetBitWhileKeyPressed under Edit bits.

Button_1			💁 Properties	🚹 Info	<b>Q</b> Diagnostics	
General						
		ŦE				
Miscellaneous	•					
Security						•
Animations		Calculation				
✓ Events		👻 Edit bits				
Click		InvertBit				
Press		ResetBit				Ξ
Release		SetBit				
Enable	Ξ	SetBitWhileKeyPressed				
Disable		ShiftAndMask				
Change		Keyboard				-

Then, click on the field Tags (input/output) and using "..." button, open the tag window. Here, it is also possible to access the interface declaration of data blocks. As tag, select auto from the Conveyor\_DB [DB1].

	ol partel partel partel partel partel	· · ·	81		- 10 KB
1212121		. 👻 🛅 HMI_1 [KTP600 PN]		Name	Ad
		🗧 🔫 🔚 HMI tags		None	
	S ENE DIE DIE DIE DIE	🗧 👻 🛅 Control conveyor [CPU 1214C	-	man	
		👻 🔂 Program blocks	-	auto	
		🕨 🕒 conveyor_DB [DB1]	-	on	
		Technological Objects	-	off	
		🕨 🔚 PLC tags	-	sensor_bottle	
			-	reset_counter	
F1	F2 F3 F4	F	-	motor	
			-	automan	
				motorauto	
Button_1					
General			-		
		<u> </u>			
Miscellaneous		Che Lasta da la			
Security	SetBitWhileKeyPressed	Add new			✓ X
Animations	Tag (Input/output)				
▼ Events	Bit	0			
Click	<add function=""></add>				

Now, the button is to flash in the automatic mode, and change color. With a double click, select under Animations\New animation Appearance.

<ul> <li>Properties</li> </ul>	Add	Animations
▼ Animations	Add	
New animation	+	Diagonal movement
Events	+	Direct movement
	•	Horizontal movement
	+	Vertical movement
	•	Appearance
	+	Visibility
	1.000	

As tag, select automan of Conveyor\_DB [DB1].

The button is to change color in the automatic mode; that means, when the variable automan has the value 1. For the color change to become visible, change the foreground color at Appearance to White and the background color to Green. At Flashing, say Yes.

		📴 Properties 🛐 Info 😨 Diagnosti					
General				-			
	Appearance						
<ul> <li>Properties</li> <li>Animations</li> <li>New animation</li> <li>Appearance</li> <li>Events</li> </ul>	Tag Name: conveyor_DB_automan Address:			Type       • Range       Multiple bits       Single bit			
	Range ▲ 1	Foreground color		ground color , 255, 31	Flashing Yes		
	4	ш				•	

Change Color of Button

Copy and paste the button Automatic. Place the inserted button under the Automatic button.

At Label, enter Manual. **Caution!** Don't press the input key; otherwise, a second line is generated.

Button_2			🔯 Properties 🐮 Info 😨 Diagnostics 🔍
General	227		
		General	
<ul> <li>Properties</li> </ul>	<b>_</b>	Mode	Label
General		Mode	Laber
Appearance		• Text	• Text Text list
Design		Graphic	Off Manual
Layout	Ξ	Graphics and text	
Text format			· · · · · · · · · · · · · · · · · · ·
Miscellaneous	20	🔘 Invisible	On Text
Security			
▼ Animations		Hotkey	
New animation		Hotkey: None	
Appearance	-	-	

HMI Tag

Under Events Press, select man from Conveyor\_DB [DB1] as tag. The variable has to be selected, because only then will the new HMI tag be generated.

HMI_1 [KTP600 PN]	Name	Ad.
🕨 🌄 Hilli tags	None	
Control conveyor [CPU 1214C	🕢 man	
🕶 👼 Program blocks	💶 auto	
<ul> <li>conveyor_DB [DB1]</li> </ul>	a on	
IEC_Counter_0	d off	
Technological Objects	sensor_bottle	
PLC tags	man     auto     auto     on     off     sensor_bottle     reset_counter     motor     automan     motorauto	
	motor	
	automan 🗧	
	motorauto	
•		

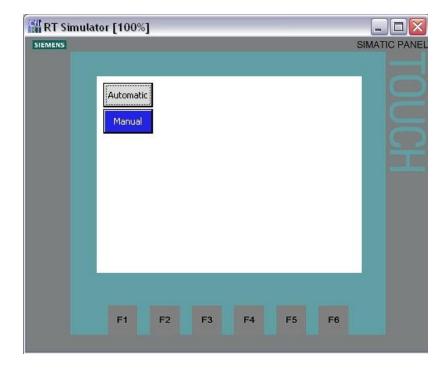
More Example HMI Tags

The button is to change color in the manual mode; that means when the variable automan has the value 0. For the color change to become visible, change the foreground color at Appearance to White and the background color to Blue. Set Flashing to No.

		Appearance _							
General	<b></b>	T			10	Trees			
Appearance		Tag				Туре			
Design		Name:	conveyor_DB_au	ıtoman	<u></u>	Range			
Layout		Address:				Multiple bi	ts		
Text format						O Single bit		\$	
Miscellaneous	Ξ.					O bingre bic	0	*	
Security		Range 🔺		Foreground color	Bac	kground color		Flashing	
		0	-	255, 255, 255		36, 36, 228	-		
New animation		<add new=""></add>					-		
Appearance		4	11	E					•
Events	-								

**Button Appearance** 

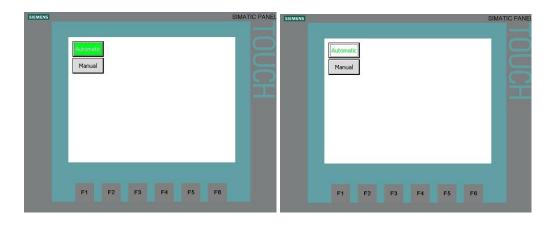
Visualization is opened in the RT simulator.



# Running the Simulator Ch 15 Human Machine Interface

Test the project of the conveyor control.

The automatic or manual mode is now pre-selected on the panel.



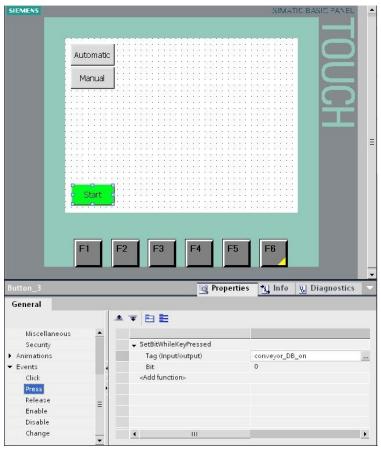
## Start and Stop Buttons

Next, we configure the start and stop buttons.

The Start button is created exactly like the automatic and manual buttons.

The Stop button has a break contact function and has to remove the signal when operated.

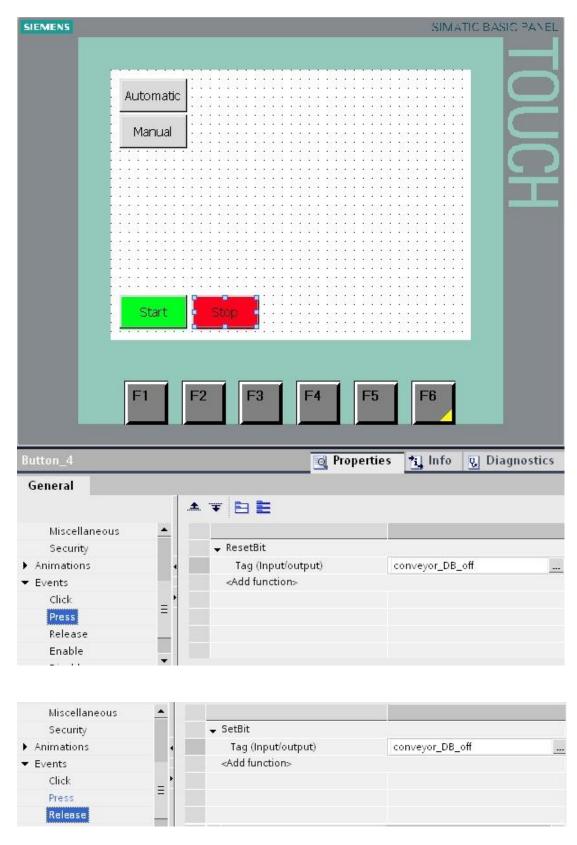
- Create the Start button
- Set the background color to Green
- Under Events Press, select under bit editing the function SetBitWhileKeyPressed.
- Then select the tag on in Conveyor\_DB [DB1].



Ch 15 Human Machine Interface

Next, do the following: Create the Stop button. Set the background color to Red.

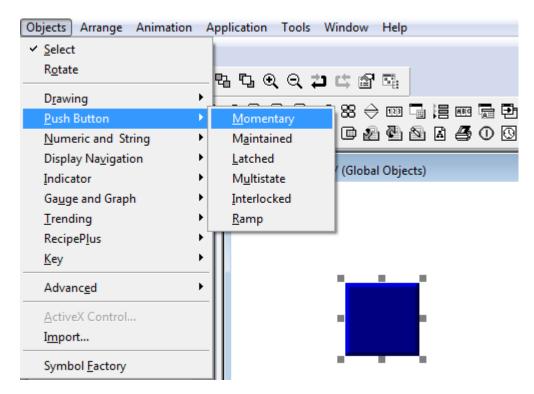
Under Events, at Press, select under Bit editing the function ResetBit and at Release the function SetBit with the tag off in Conveyor\_DB [DB1].



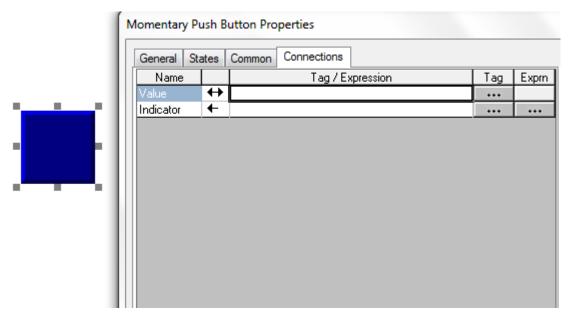
Screen with Start and Stop Buttons

## **Allen-Bradley Button Configuration**

The choice of button type indicates the type of function desired. There is no need to program both press and reset but rather only type and the function is performed.



Choice of Button Type Determines Function



Connections Tab on Push Button Function

The tag or expression may be programmed for the value as well as for an indicator. The indicator actually is a second button with the state desiring to be displayed. This value may be the same address as the value entry or a second address. To perform a similar function, Siemens may require two buttons overlaying each other.

Before we test the visualization, first another change has to be made in the Step7 program In OB1, remove the assignment S3 and S4 when calling FB1.

Save and load the modified program.

SIEMENS							SIMATIC PAN	EL
	Automatic						UUCH	
	Start	Sto	p					
	F1	F2	F3	F4	F5	F6		

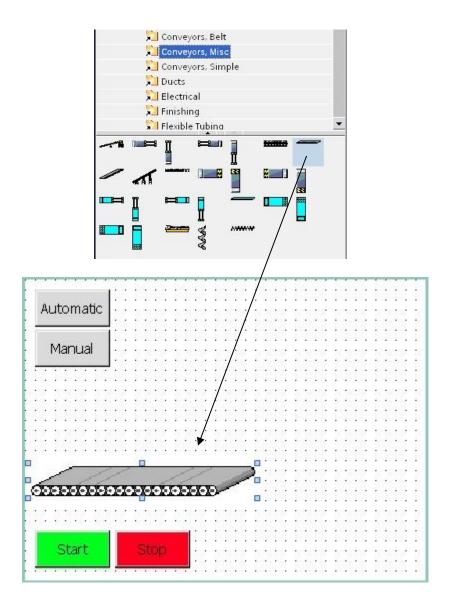
Load the configuration to the panel, and test the Start and Stop buttons.

## Inserting Graphics from the Graphics Folder

In the tool box under Graphics, open the directory tree WinCC graphics folder\SymbolFactory 256Colors\Conveyors, Misc.

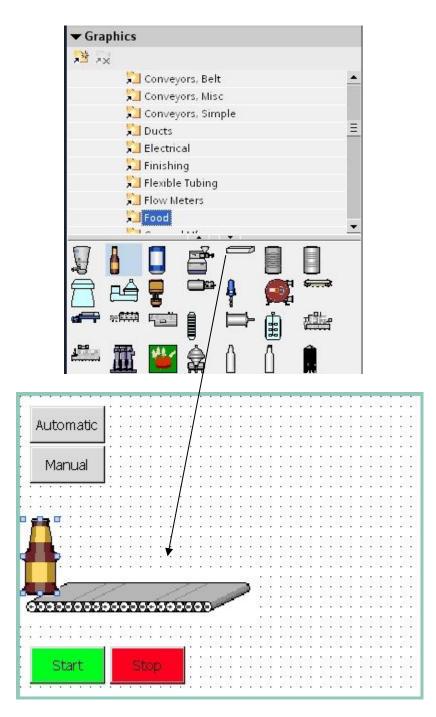
Drag and drop the graphic of the conveyor belt to the root screen.

▼ Graphics	
× 1	
← 🔁 WinCC graphics folder	
🗢 知 Symbol Factory Graphics	
🗢 📜 SymbolFactory 256 Colors	=
🟓 3-D Pushbuttons Etc	
📜 Air Conditioning	
📜 Architectural	
知 Arrows	
📜 ASHRAE Controls & Equipment	



In the tool box under Graphics, open the directory tree WinCC Graphic folder\SymbolFactory 256 Colors\Food.

Then drag and drop the picture of the beer bottle in the root screen. Change the size and the position of the bottle.



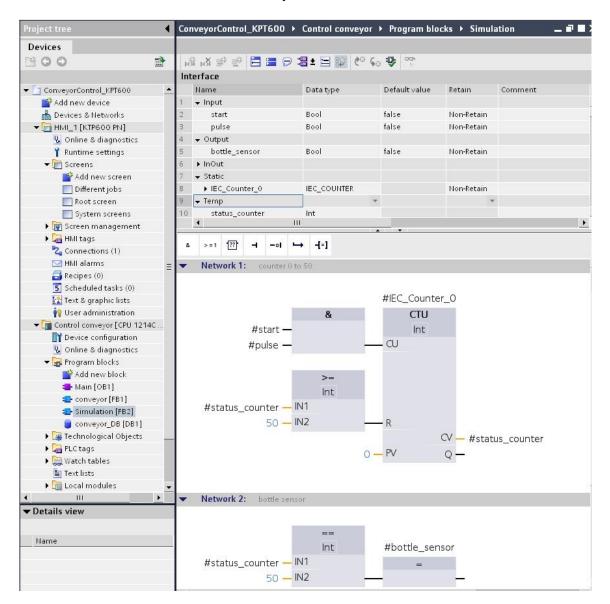
# Note

All screen objects have to be within the work area (320x240 pixels).

## **Control Program for Simulating the Bottle Movement**

To simulate the bottle movement and the bottle sensor, we create a new block. The FB2 (simulation) below with tag declaration and networks consists of a counter that, through a start signal, always counts up from 0 to 50.

In Network 1, the CTU (count upward) is inserted as multi-instance. In Network 2, a bottle sensor pulse signal is read out when the count 50 is reached. This simulates when a bottle leaves the conveyor.



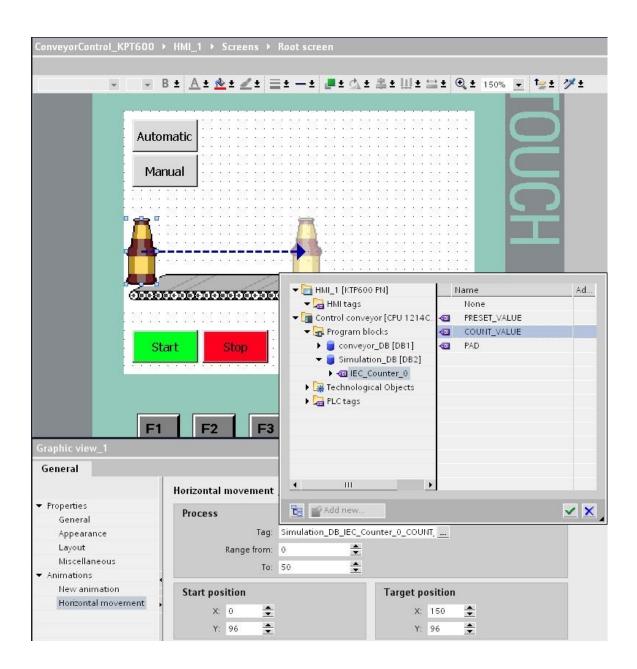
# **Configuring the Bottle Movement**

Highlight the bottle and select under the tab Properties at Animations - Horizontal movement (double click).

ConveyorControl_KF	т600 → нмі_1 →	Screens	► Roo	t screer	ı			_ •	∎×
٧	- B ± <u>A</u> ±	枪 ± <u>/</u> ±	≣± ·	-± 🛃	± 🗛	± 🛎 ± I	ll ± ≌ ±	🔍 ±	•
									<b>^</b>
	Automatic								
	Automatic								
	Manual								
	<b></b>								
	<b>11</b>								=
									=
			/ /						
	~~~~			and the second second					
	00000000000	000000	1000						
	Start	Stop	1.1.1						
· · · · ·		otop							
									_
			_						
4			-						• •
Graphic view_1				Proper	ties	1 Info	😨 Diag	nostic	
				Hope	acs	<u>1</u>	To Drug	no suc.	
General									
<ul> <li>Properties</li> </ul>									
General	Add	Animatio							
Appearance	+	Diagonal							
Layout		Direct me							
Miscellaneous	+	Horizonta							
<ul> <li>Animations</li> </ul>	+	Vertical r		nt					
New animation	+	Appeara	nce						
	+	Visibility							

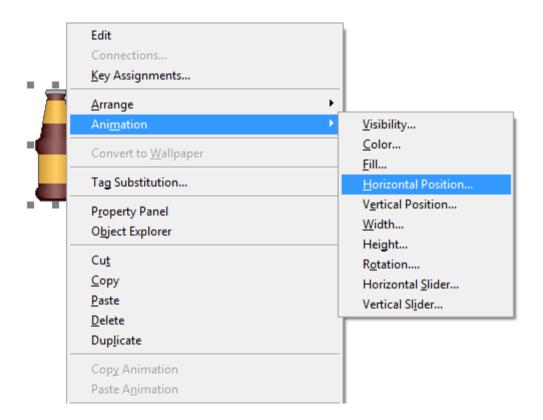
As variable, select COUNT\_VALUE of the Simulation\_DB (DB2). For range, enter from 0 to 50.

Change the bottle's target position up to the conveyor end X150.



#### **Allen-Bradley Animation**

The Allen-Bradley animation proceeds much the same as Siemens in that the device edited may be animated in a number of ways including horizontal position. The position is a function of a number in a location which is monitored and the beer bottle moved accordingly.



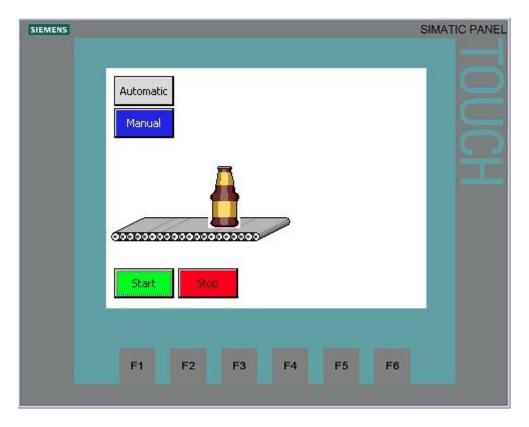
Eil	Touch	Color	0LE Ver
⊻isibility	<u>R</u> otation	∐ <u>W</u> idth	∬ Height
Horizontal Position	Vertical Position	Horizontal Slider	Vertical Slider
Expression range		Horizontal offset (Pixels) At minimum:	At maximum: 🔲 0
Use tag's min and max pr	operty values	At minimum:	At maximum:
Constant Mir:	L		
	Max		

In the project window, select the HMI tags.

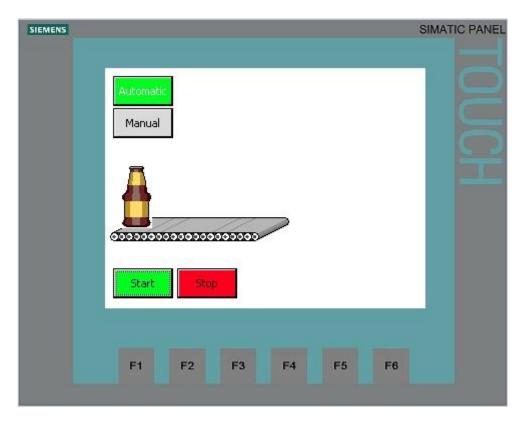
Project tree	•
Devices	
B 0 0	<b>₫</b>
] ConveyorControl_KPT600	
📑 Add new device	
🚠 Devices & Networks	
🛅 HMI_1 [KTP600 PN]	
🛂 Online & diagnostics	
🍸 Runtime settings	
🕶 🛅 Screens	
💕 Add new screen	
📃 Different jobs	
🔲 Root screen	
System screens	
🕨 🙀 Screen management	
🕶 🔙 HMI tags	
🛥 HMI tags (6)	
🍡 Connections (1)	
🖂 HMI alarms	
🛃 Recipes (0)	
5 Scheduled tasks (0)	
🔛 Text & graphic lists	
🛊 💡 User administration	

					📹 HMI tags	🖂 HMI alarms
MI ta	gs					
	Name 🔺	Data type	PLC tag	Address	Array elements	Acquisition cycle
-00	conveyor_DB_auto	Bool	conveyor_DB.auto	<symbolic access=""></symbolic>	1	100 ms
-	conveyor_DB_automan	Bool	conveyor_DB.automan	<symbolic access=""></symbolic>	1	100 ms
-	conveyor_DB_man	Bool	conveyor_DB.man	<symbolic access=""></symbolic>	1	100 ms
-	conveyor_DB_off	Bool	conveyor_DB.off	<symbolic access=""></symbolic>	1	100 ms
-00	conveyor_DB_on	Bool	conveyor_DB.on	<symbolic access=""></symbolic>	1	100 ms
-	Simulation_DB_IEC_Counter_0	Int	Simulation_DB.IEC_Coun	<symbolic access=""></symbolic>	1	100 ms
	<add new=""></add>					<b>A</b>

Drag the slider in the window to the right in order to get to the column Acquisition cycle. Set the acquisition cycle of the HMI tag to 100ms.



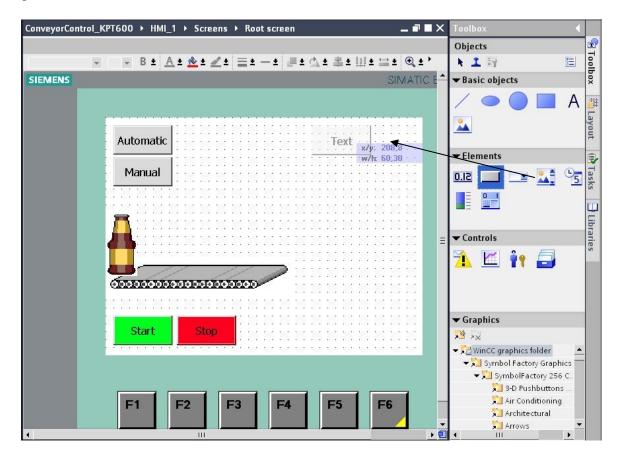
Then save the project, load it to the panel and test it.



After 20 bottles, the conveyor motor stops. The bottle counter has to be reset before the next start.

#### **Resetting the Bottle Counter**

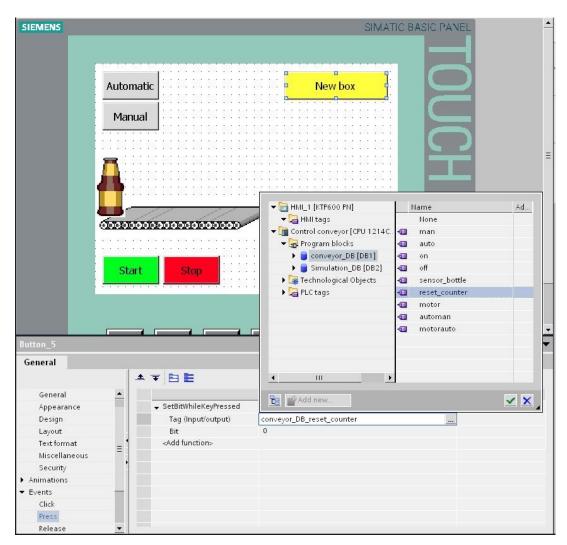
Drag a button into the root screen.



As text, enter Exchange beer case and adjust the color Position & size to the button.

General	
Mode	
Text     Graphic     Graphics and text     Invisible  Hotkey Hotkey: None	Label • Text Text list Off Hew box
Layout Position & size X: 200 ◆ ₩ 110 ◆ Y: 8 ◆ I 30 ◆	Fit to size Fit object to contents
	Graphic Graphics and text Invisible Hotkey Hotkey: None ▼

Under Events Press, select under bit editing the function SetBitWhileKeyPressed. Select the tag reset\_counter from Conveyor\_DB [DB1].



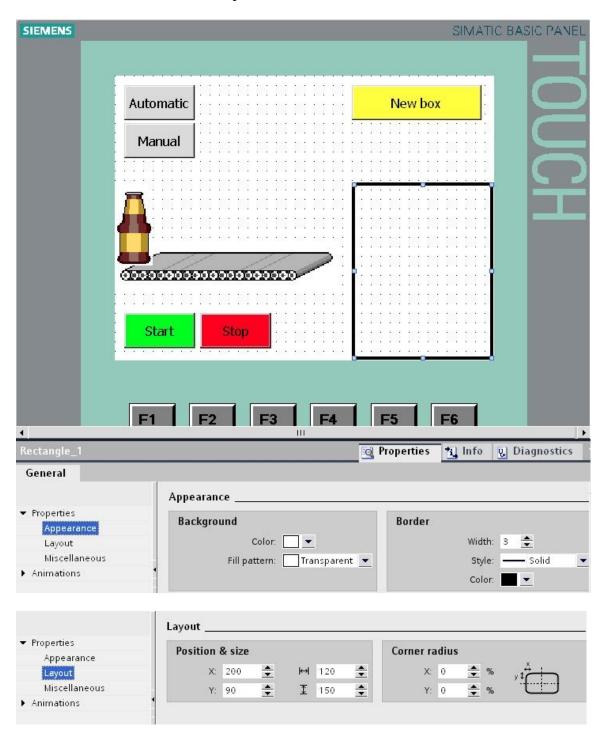
Set the acquisition cycle of the new HMI tag to 100ms. Then save the project, load it to the panel and test it.

SIEMENS			SIMATIC PANEL
Automa		New box	
			Ŧ
Start	Stop		
F1	F2 F3 F4	F5 F6	

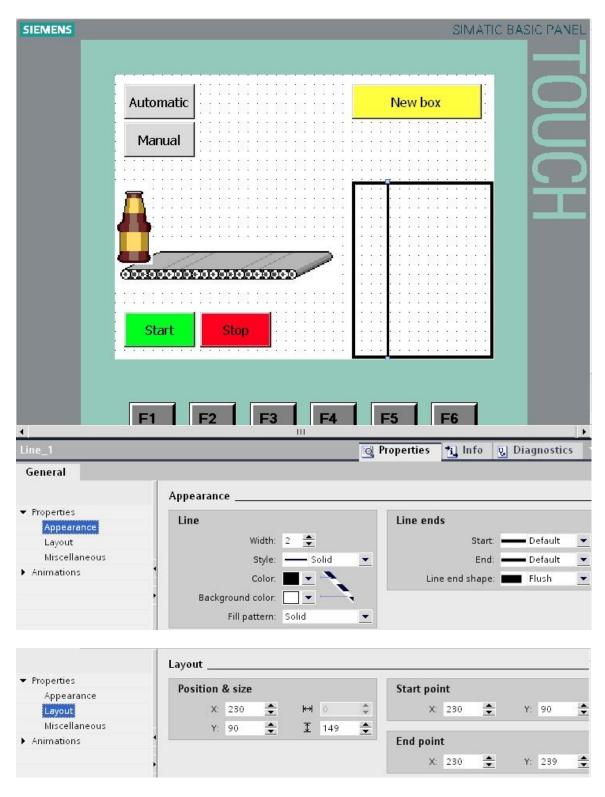
Ch 15 Human Machine Interface

#### **Drawing the Beer Case**

Draw a rectangle with a transparent background. Enter the width of the border, the position and size.



Draw a vertical line at a distance of 30 pixels.



#### Note

Although the measurements of the line are correct, it is drawn beyond the rectangle. Change the form of the line ends to flush, and shorten the line by one pixel from 150 to 149. Draw a horizontal line spaced at 30 pixels

SIEMENS		SIMATIC BASIC PANEL
	matic nual	New box
↓ Line_2 General		F5 F6
<ul> <li>Properties         Appearance         Layout         Miscellaneous         Animations     </li> </ul>	Appearance	Line ends Start: Default End: Default Line end shape: Flush
<ul> <li>Properties         Appearance         Layout         Miscellaneous         Animations           </li> </ul>	Layout	Start point         Y: 120         ↓           X: 200         ↓         Y: 120         ↓           End point         X: 319         ↓         Y: 120         ↓

With copy and paste, create the remaining lines spaced at 30 pixels.

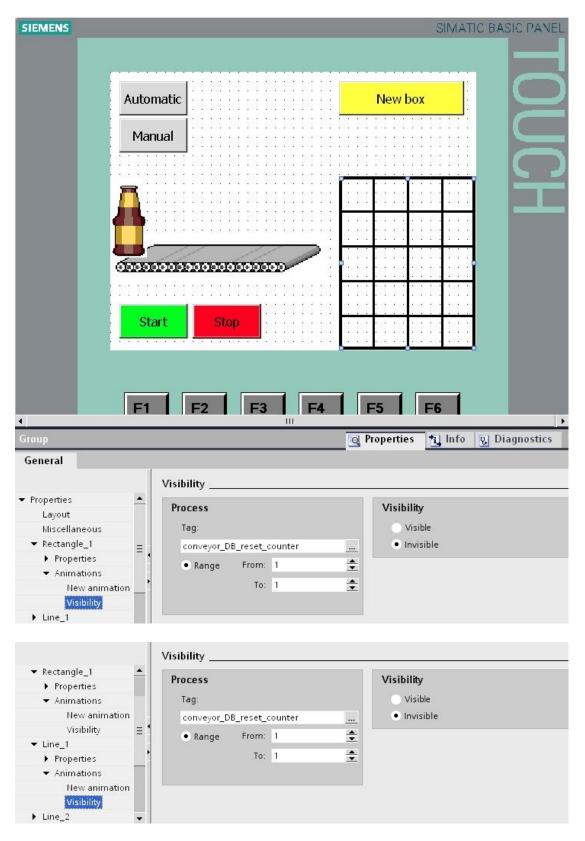
Highlight the beer case by dragging a border around it with the mouse.

Automatic					Ne	w box
						· · · · · · · · · ·
Manual						
	:::::	111				
					••••	· · · · · · · · · · · · · · · · · · ·
						: ::::::::::::::::::::::::::::::::::::
					<u></u>	<u></u>
					∦	
	1		1	- Ca		<u> </u>
		~~~		- ·	🦹 .	
	000000					
					1	: : <b>:</b> : : : : : : : : : : : : : : : :
10000 10000 10000 1	AN 3533 3543			. in		
Start	Sto	D		111		
	1000		1.21 1.21	1.21 E	· · · ∦ · ·	

In the menu Edit select the function Group

Projec	: Edit View Insert On	line Options Tools W
Br B	Open object	i 🖬 🗙 🌆 🖥 🛄
Pro	Cut Ctrl+X	ConveyorCon
	X Delete Del Rename F2	
Visualization	Alignment  Placement  Size  Rotate  Flip	
	Order  Group	fills but the
	Layers 🕨	Remove from group
	Tab sequence 🕨	Ungroup
	Select all Ctrl+A	Edit group
	🙀 Find and replace Ctrl+F	
	Compile Ctrl+B	=
	Know-how protection 🕨	
	Q Properties Alt+Enter	

We don't want to display the rectangle and the lines when the beer case is exchanged. At Rectangle\_1 and at the lines, generate the animation Visibility using the tag Conveyor\_DB\_reset\_counter at value 1 Invisible. For the lines, the animation can also be copy/pasted.



Then save the project, load it to the panel and test it.

# Drawing Bottles in the Case

Miscellaneous

Animations

Enlarge the view and draw a circle in the lower right field of the box.

					ot screen					_ 1
			1 4 1 4	<i>a</i> . [= .			* <b>.</b> 111 <b>.</b> 4	→		1 <b>1</b>
	¥ ¥		. ± <u>×</u> ± 4		- I [	I T CV T I	≝±Ш±:		<b>2 800%</b>	
55	15						10	15		
85	12	83	53	53		100	53	12	12	
17	10	87	83	52		102	83			
17		82	83	5		12	83	0		
		4				4	•			
85	15	15	15	53	15	12				
			20.0							
			25	10	10	122				
12		52								
5.2		333								
80 80	12	12 X	et.	ea.	87					
		82 87			85					
2	17	87	2	2	2	83		<b>.</b>	_	
	2				2			_		
rcle_1	12	27 27		-	2	in the second se	10	1 Info		
						Pro	10	*ij Info	<u>ए.</u> Diagn	
		Арре	arance		25	Pro	111	*i] Info		
General Properties		_	arance				III operties	1 Info		
General Properties Appearance	e	_	arance	Color:		Pro	111		ए, Diagn	
ircle_1 General Properties Appearance Layout Miscellaned		_	kground	Color:		Pro	III operties	Width		ostics

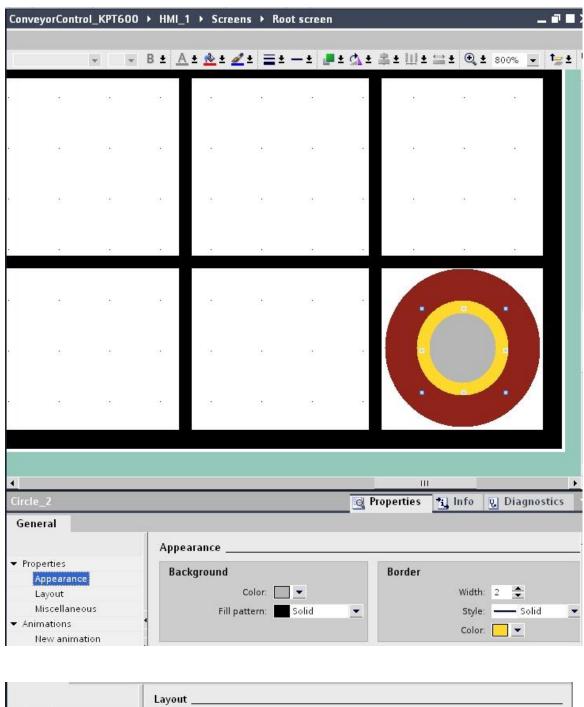
-

**1** 24

\*

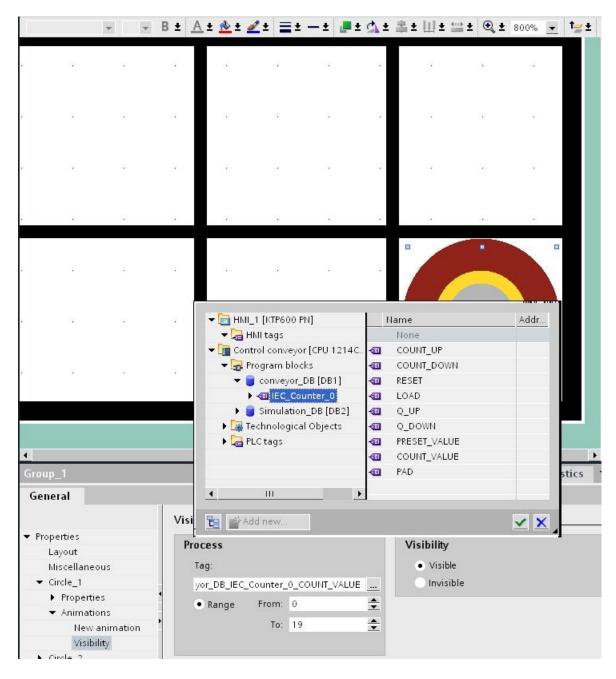
Y: 212

Draw a second circle.

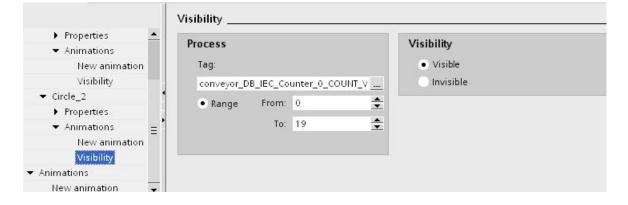


<ul> <li>Properties</li> </ul>	Position	0 0170				3.	Radius		
Appearance	Position	& size					nadius		
Layout	X:	297	-	₩	14	*	Radius:	7	-
Miscellaneous	Y:	217	÷	Ŧ	14	*			
Animations	•	201202011							
New animation									

Group the two inserted cycles.

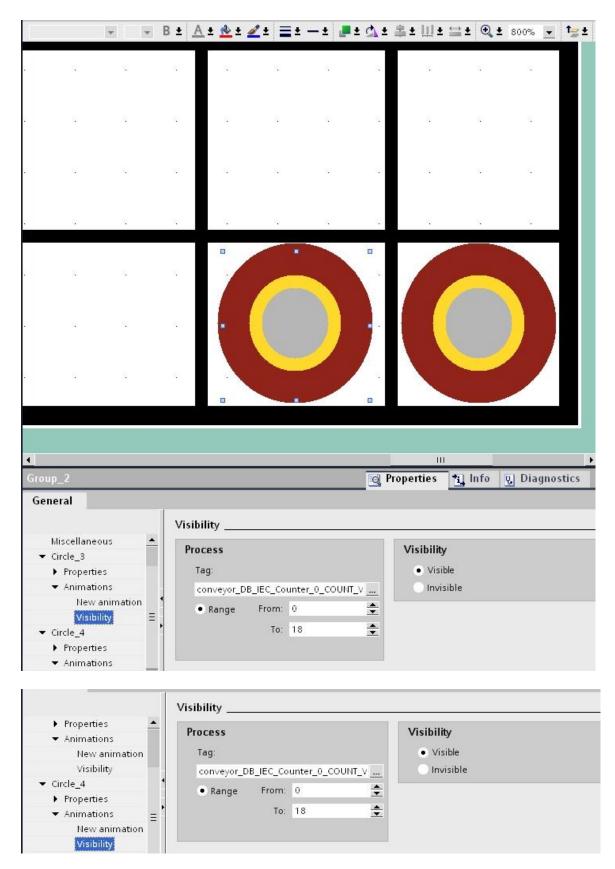


At Circle\_1 and Circle\_2, generate the animation Visibility with the tag Conveyor\_DB\_IEC\_Counter\_0\_COUNT\_VALUE value range 0 to19 Visible.



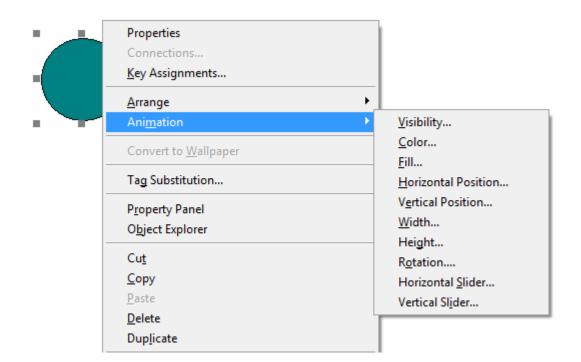
Copy and paste the bottle.

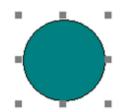
For both circles, under Visibility change the value range of the tag Conveyor\_DB\_IEC\_Counter\_0\_COUNT\_VALUE to 0 to18 Visible.



Ch 15 Human Machine Interface

Animation with visibility is available with Allen-Bradley as well. It is shown below with visibility as a property with a tag or an expression with tags available to provide logic for the visibility of a device, in this case, a circle.





Horizontal Position	Vertical Position	Horizontal Slider	Vertical Slider
<u> </u>	Touch	Color	0 <u>L</u> E Verb
⊻isibility	<u>R</u> otation	<u>₩</u> idth	Height
Expression			
			A Tags
			+ Expression
Expression true state			
<ul> <li>Invisible</li> <li>Visible</li> </ul>			

The expression editor gives the ability of the programmer to add logic to select the visibility of an object.

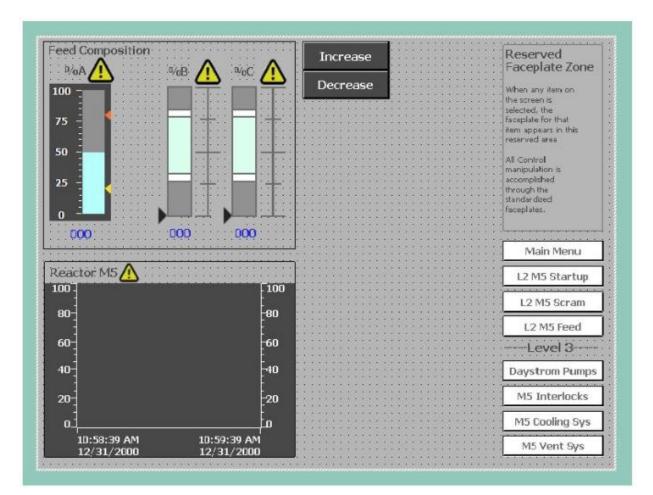
ression Editor Expression	-
A	+
If     Logical     Relational     Arithmetic     Bitwise     Functions     T       Check     Syntax	ags
ine: 1 Column: 1	
OK Cancel	Help

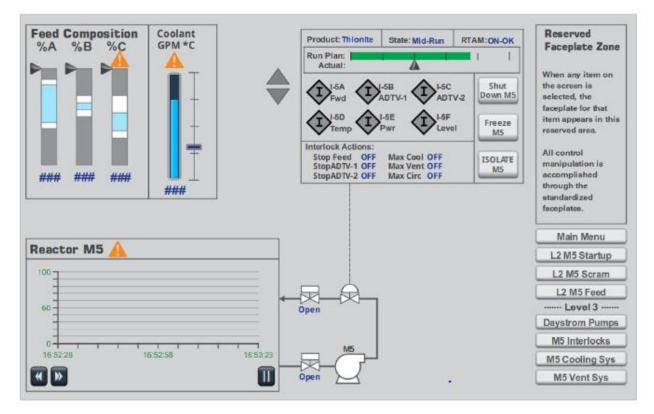
Copy and paste the individual bottles.

At the animation Visibility of both circles decrease the value to by 1. The last bottle has the value range from 0 to 0.

Automatic	New box	
Manual		
R		
<b></b>	<u> </u>	
6000000		
Start		

The following are two graphic examples created by EET student Lee Monday. They give examples of the type of HMI screens discussed in the next section on HPHMI.





Ch 15 Human Machine Interface

# How to Improve Plant Operations through Better HMI Graphics

A number of good texts are available for better design of HMI graphics including:

**The High Performance HMI Handbook** by Bill Hollifield, Dana Oliver, Ian Nimmo and Eddie Habibi

**Designing for Situation Awareness (An approach to User-centered Design)** by Mica R. Endsley and Debra G. Jones

Human Machine Interface (HMI) Design: The Good, The Bad, and The Ugly (and what makes them so) by Paul Gruhn, P.E.

From Hollifield: "Five areas are primary in the successful design of a good HMI screen system. They are:

- Situation Awareness
- Using Color Effectively
- Interpreting the data
- Depicting Device State
- HMI Display Organization

Situation awareness relates to the goals and objectives of a specific job or function. First of all, designers and engineers form in their heads another mental model of the process than an operator. By understanding how operators select and use goals, designers can better understand how information is perceived. Without understanding the user's goals on Situation Awareness, the information that is presented has no meaning."

Hollifield gives a number of graphic displays of the evolution of the operator interface. The following figure was found until the mid 1980's. These panels were created and manufactured for the plant floor and were extremely difficult to change once deployed.



Fig. 15-17 Example of a Control Wall

More computer driven HMI panels began emerging to replace these panels although the early computer replacements were extremely expensive, especially when compared to today's HMI. Early HMI panels also had the disadvantage of little ability to extract or analyze process data. Ch 15 Human Machine Interface From Hollifield: "When the modern systems were introduced, they included the capability to create and display graphics for aiding in the control of the operation. However, there were no guidelines available as to how to create *effective* graphics. Early adopters created graphics that mimicked P&ID or schematic drawings, primarily because they were readily available. The limited color palette was used inconsistently, and screens began to be little more than crowded displays of numbers on a P&ID."

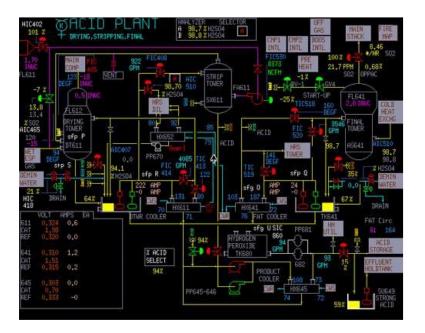


Fig. 15-18 Early Graphic Showing Many Problematic Practices

Early graphics were a stark contrast to the hard-wired panels of earlier times. They were developed primarily from the P&ID drawings of the process but lacked any methods to quickly diagnose problems in the plant. While this style was accepted and promoted, there are severe problems with this design as noted by Hollifield and others. The next few pages outline some of the common mistakes made by continuing to use the design shown above.

With each design, the operator must be considered first. The operator may be a person with little or no process education and may have little or no training. He or she may be on the job only a few days and be required to make quick decisions concerning the operation of the plant. While there are not many examples of plants being destroyed because of operator error, the operation of the plant must be considered from all angles and the HMI panel is the first place an operator or maintenance person should go for help. The following example shows 3-D design but is of little help with actual information.

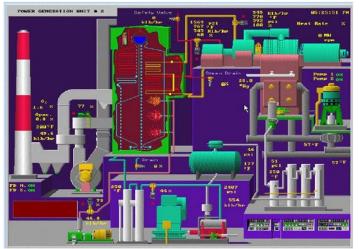


Fig. 15-19 A Flashy Graphic inappropriate for Actual Operational Control

Ch 15 Human Machine Interface

Hollifield points out: "There are no trends, condition indicators, or key performance elements. You cannot easily tell from this graphic whether the operation is running well or poorly. That situation is true for more than 90 percent of the graphics used throughout industry today because they were not designed to incorporate such information. Instead, they simply display dozens to hundreds of raw numbers lacking any informative context."

## **Justification for HMI Improvement**

An industry with a great interest in excellent HMI displays is the avionics industry. The following is a flight simulator for a small plane. This training is of vital importance prior to the pilot student getting into a plane.



Fig. 15-20 Garmin G2000 Avionics Package in a Small Plane

The following shows an example of a great deal of information yet no real information about the process. This is not what to do!

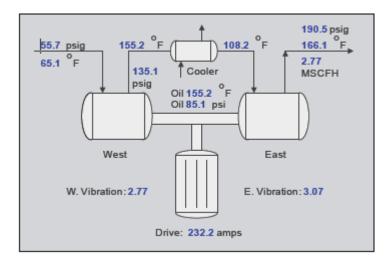


Fig. 15-21 All Data, No Information

The same information as depicted in Fig. 15-21 above is displayed in the graphic below in Fug, 15-22. It shows data pertinent to the process with enough information about those values to make logical judgments about the process. The data is mostly analog.

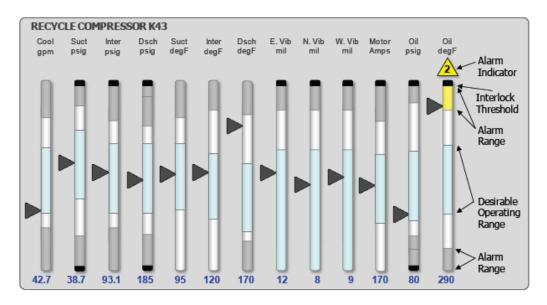
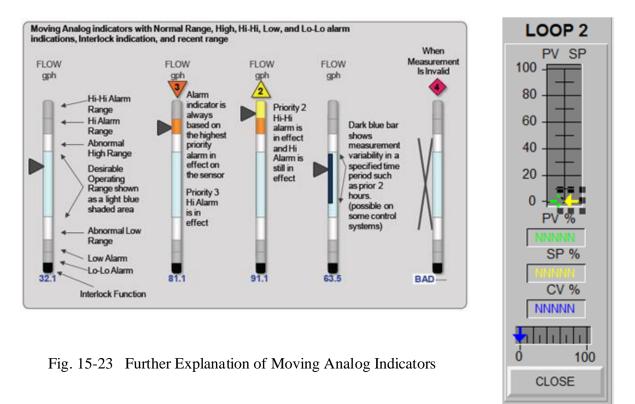


Fig. 15-22 Analog Depiction of Information



In the figure above, PID controllers are shown with active values displayed. Pointers show the actual value on a sliding scale. Alarms are shown with color and verbiage. The PID faceplate at right is the more classical PID graphic. Either type of display is acceptable but fundamental values are necessary in either. The written warnings at left are desired with either display.

# From Hollifield again, "Color, by itself, is never used as the sole differentiator of an important condition or status

Most graphics throughout the world violate this principle. A color palette must have a limited number of distinguishable colors used consistently. Bright colors are primarily used to bring or draw attention to abnormal situations, not normal ones. Screens depicting the operation running

normally should not show brightly saturated colors, such as bright red or green pumps, equipment, valves, and similar items."

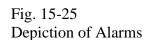
Tables can be useful for display of large amounts of data. The HMI display should be used to instruct or teach the operator as changes in the process necessitate action:

Display of process v	alues. No ala	arm condi	tions are in e	effect.
480.1 psi	BAD	psi	OOR	psi
Normal	Bad	Value	Out of	Range
Note that the BAD or additional alarm elem conditions).			,	
It may be possible to measurement, rather the depiction will be:				
	?LG 480.	1 psi		
1	Bad or Out o	of Range	Value	
Discrete ("digital") cor color and in bold, usi			words	
	amper 7: oving	SW 60		/ent Valve Auto Closed
Selection of an item of and the selected item				
480.1 psi		480.1	psi	
Invisible selection a	rea	Selection	n indication	,

A good example of depiction of alarms is the following:

Poor Alarm Depict	ions Violating the	Basic Color Rule		
480.1 psi No Alarm	480.1 psi In Alarm	480.1 psi In Alarm	Similar Bad Depictions	480.1 480.1 480.1
480.1 psi Diagnostic Priority	480.1 psi Priority 3	480.1 ps Priority 2	i 480.1 Priori	p or
480.1 psi Diagnostic Priority	480.1 psi Priority 3	480.1 ps Priority 2	i 480.1 Priori	psi ty 1
Improved Alarm De	epictions With Red	undant Coding	si 480	.1 psi
Priority 4 Diagnostic	Priority 3	Priority 2	Pri	ority 1
Suppressed S	480.1 psi	a S t	he symbol's ou bove are impo ymbols withou hem would not	rtant! t : be as
	s Summary ele		ffective in stan	
Alarms 1 Total 0		4 S 1 12	4 3	
Un-Ack 0	0 2	1	2 1	

Fig. 15-24 Depicting Process Values



Ch 15 Human Machine Interface

For linked alarms, the following is a good example:

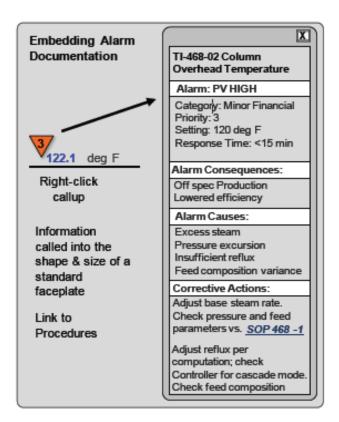
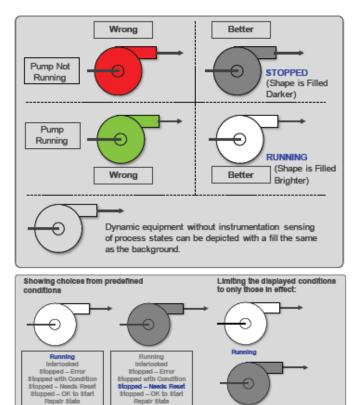


Fig. 15-26 Linked Alarm Information

# **Depicting Dynamic Equipment**

Use of color for running conditions of motors is not good. The following shows the better use of color brightness for running vs stopped motors.



The relative brightness of the object shows its "ON-oFF" status, as does the use of a process value word next to it. Equipment items brighter than the background are "ON" (think of a light bulb inside them). Items darker than the background are "OFF." If equipment has no status that is sensed by the control system, but is desired on the graphic anyway, it is shown as transparent to the background color. The status word can indicate several conditions, as shown. Remember, if any of those are also alarm conditions, the separate alarm indicator will appear next to the equipment when it is in an alarmed state.

Fig. 15-27 - Depicting Status with Redundant Coding and Proper Color Usage

Ch 15 Human Machine Interface

#### **Bars vs. Pointers**

Pointers are better for showing analog values rather than bar graphs. The following point out this advantage.

	Anal	Analog Position – Poor but Typical					Analog Position - Better				
	0%	25%	50%	75%	100%		0%	25%	50%	75%	100%
alve						Valve					
V-1					100%	V-1	_				_ 1
V-2					100%	V-2					
V-3					95%	V-3					
X-1					88%	X-1					
X-2					100%	X-2					<b></b> 1
X-3					100%	X-3					1
X-4					75%	X-4					
S-1					0%	S-1				_	
S-2					55%	S-2				_	
K-1					100%	K-1				_	<b></b> 1
K-2					100%	K-2				_	= 1

Fig. 15-28 Bars vs Pointers

#### **Depicting Level Indication**

The following show a progression from very poor to very good depiction of level control in a tank.

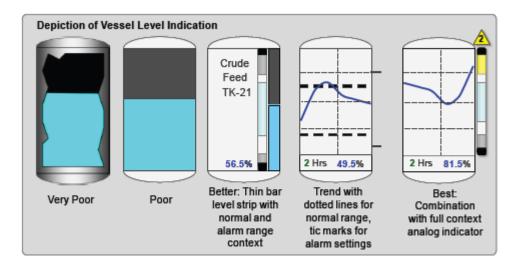
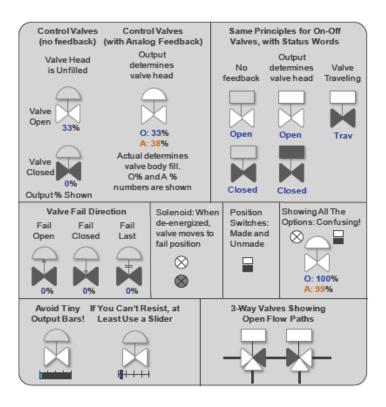
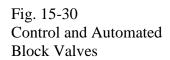


Fig. 15-29 Vessel Levels

### **Depicting Control Valves and Shutoff Valves**

The following gives examples of control valves. Closed position is shown in black and open in white





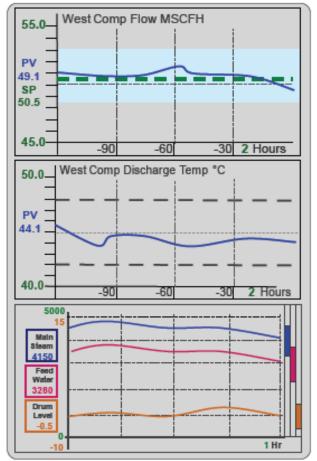


Fig. 15-31 Trend Depiction of Desirable Ranges

Trends should be **embedded** in the graphics and appear, showing proper history, whenever the graphic is called up. This is generally possible but is a capability often not utilized. Trends should incorporate elements that depict both the normal and abnormal ranges for the trended value

Ch 15 Human Machine Interface

Table Desig	gn				
nproper P	ractice Be	tter Prac	tice	HPHMI Startup Permis	sives Table
Abc abc a		abc abc	Breaker 15 Power	OFF	
Abc abc a		abc abc	Oil Temp 16-33	NOT OK	
Abc abc a	7100	abc abc	Oil Pres Status	NOT OK	
Abc abc a	abc Abc	abc abc	Level in TK-8776	NOT OK	
				Gen System Status	OFF
HPHMIE	quipment Sta	itus Tabl	e	Comp 88 in Auto	OK
Air	Status	Mode	Diagnostic	Lineup Ready	NOT OK
Comp				Sys Status Checks	NOT OK
C#1	RUNNING	AUTO	ок	Bearing Readouts	OK
				Comm check	NOT OK
C #2	STOPPED	MAN	ок	Outlet Temp < 250	NOT OK
C #3	RUNNING	AUTO	ок	Cooling Flow	OK
C#4	STOPPED	AUTO	FAULT 3	Internal Circuit Check	NOT OK
6 #4	STOFFED	AUTO		Bypass Closed	NOT OK
Δnother	Better Practi	ce Status	Table	AFS Function	NOT OK
Anounci	Pump		JIUDIC		
A2 CWP	A2 HWP ON	C2 HWP ON	A2BFPT ON		
B2 CWP	B2 HWP OFF	SUBFP ON	B2BFPT ON		

Fig. 15-32 Tables and Checklists

#### **Depicting Shutdown Activation**

"Operators must have the ability to shut down operating equipment manually and quickly. However, when an important action with significant consequences is based upon operator input, the input should have a confirmation mechanism that avoids inadvertent activation. The "cancellation" option should be consistently implemented.

It should never be possible to make a single selection on a screen that results in an inadvertent shutdown. A "Shutdown button" should call up at least one, and perhaps two, layers of confirmation before it is possible to actually cause such a significant event."

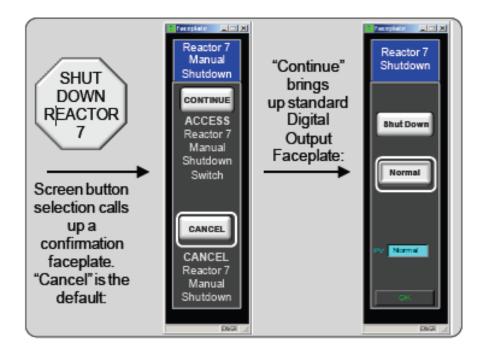


Fig. 15-33 Layers of Confirmation

## **Depicting Interlock Functionality**

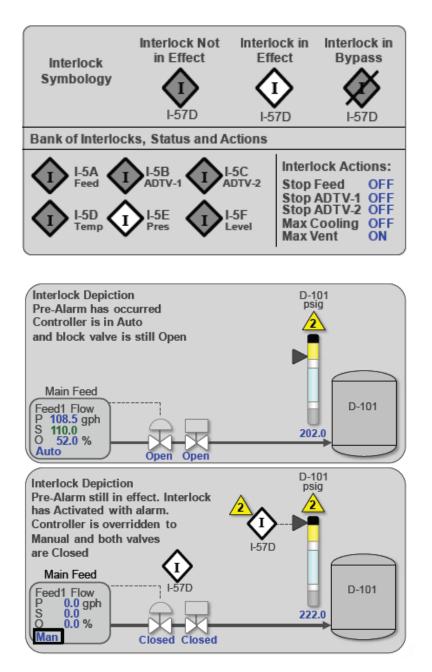


Fig. 15-34 Interlock Symbology Example

Fig. 15-35 Interlock Before and After Activation

West Compressor Interloci	k Initiators	Mech W-1B Flow	I W-1C Pres
Comp in Overspeed	NO	1 Stg Pres is High	YES
Winding Temp is High	NO	2 Stg Pres is High	YES
Vibration is High	NO	Suction Pres is Low	NO
Oil Pres is Low	NO	Total Flow is Low	NO
West Compressor Interloc	k Actions		
W. Comp Shutdown	NO	Flow Cascade to Manual	YES
Inlet Block Valve	OPEN	Maximum Flow Bypass	OPEN
Outlet Block Valve	OPEN	Maximum Cooling	NORM
E. Comp Override to 100%	NO	Winding Purge	OFF

Fig. 15-36 Interlock Diagnostic Table

Compressor 43 Shutdown	Initi	ators	
Overspeed		ОК	
West VIB >25	х	1.5	First Out
West Bearing Temp >220		160.3	
East VIB >25		1.3	
East Bearing Temp >220		155.3	
Suction Pressure <20	х	12.3	
Discharge Temp >240		148.7	
Oil Pressure <10	Х	3.3	

Fig. 15-37 Shutdown Initiator with First-Out

# Startup Map

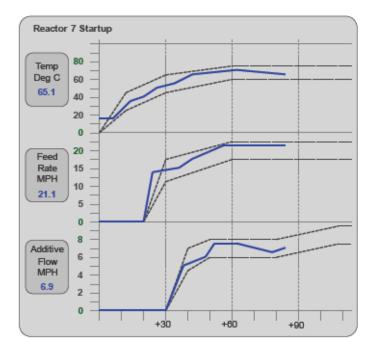


Fig. 15-38 High Performance Element Designed for Startup Use

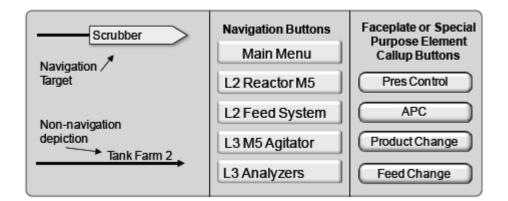


Fig. 15-39 Navigation Buttons and Targets

## **Display Hierarchy**

Displays should be designed in a hierarchy that provides progressive exposure of detail.

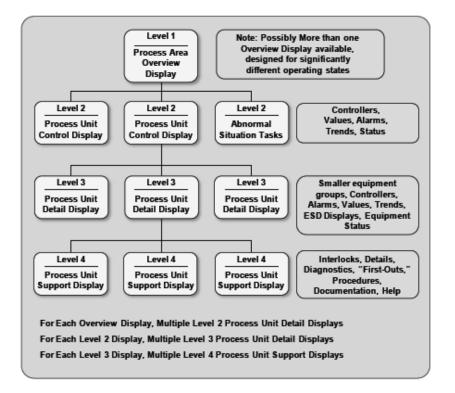
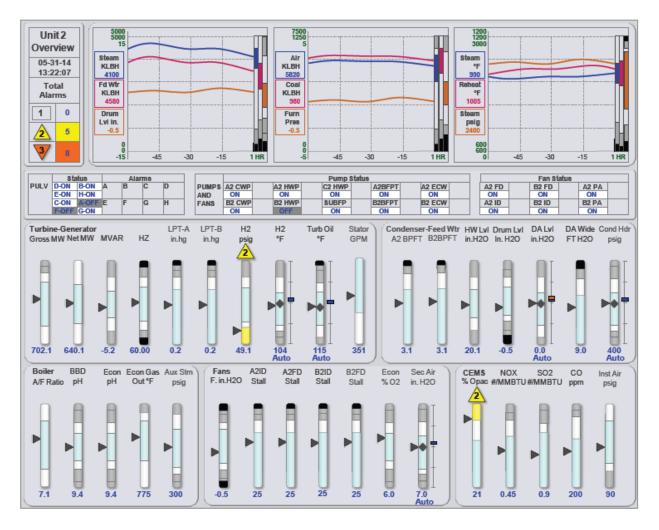
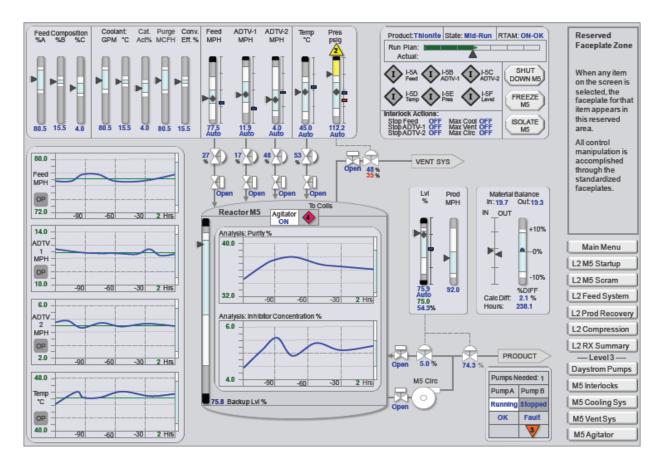


Fig. 15-40 High Performance HMI Display Hierarchy

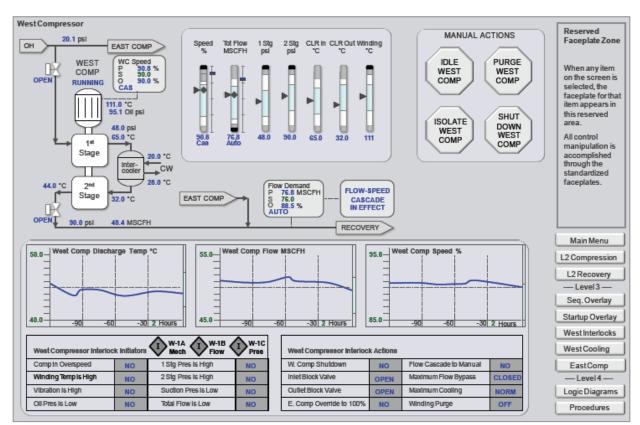
# Level 1 – Operation Overview



Level 2 – Unit Control



Level 3 – Unit Detail



(Level 4 is used for support and diagnostic displays.) Ch 15 Human Machine Interface

### **OPC and Visual Basic**

OPC is short for OLE for Process Control. OLE is short for Object Linking and Embedding. OPC strives to connect industrial automation with software programs (sometimes referred to as enterprise systems) to share data. OPC is an open system with shared standard approaches. Currently seven standards comprise the OPC system. OPC Foundation is the organization to oversee the adoption and creation of these standards.

OPC is a program that works with OLE (Object Linking and Embedding) a technology developed by Microsoft for the purpose of embedding and linking to documents. OLE stands for OLE for Process Control. Included in OPC are devices that provide aid to the programmer in areas such as:

- Alarms and Event
- **Redundancy:** Industrial applications frequently require high availability and reliability that can be easily achieved by implementing communication redundancy
- **Client Server Architecture:** The client/server nature of OPC enables users to architect connectivity solutions that would previously be prohibitively expensive.
- **Historical Data Access:** OPC Historical Data Access (OPC HDA) specification is used to archive and retrieve process data. Also included are trends reports using the OPC HDA client applications.

DDE and OPC are integrated into the Allen-Bradley product through RSLinx. See the opening tabs for these applications in RSLinx below:

Connect	Machine Name:	Server Name:	Version Info:		
OPC Group Configu	uration	OPC Grou	ıp Data		
Lopic:	Item :	Value:	ale (msec):		Fig. 15-4
OPC Group function Async Read OPC Item functions	Async Write Asy	nc Refresh Advise	al updates:	Sync Write	

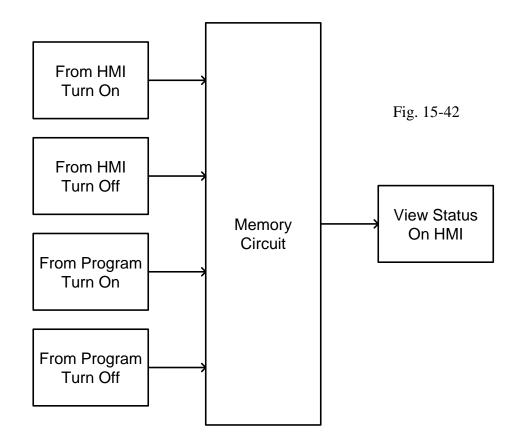
Microsoft's Access and Crystal Reports are examples of the power of using OPC with Visual Basic.

#### **Graphic Standards From Windows Standards**

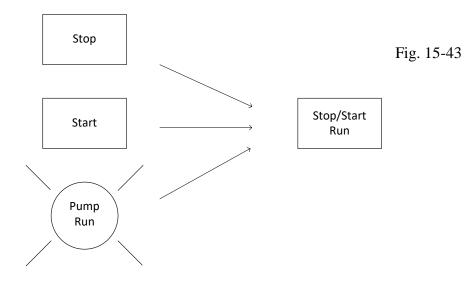
Graphics come in many flavors but not all file formats are suitable for all purposes. How do you know which is best? Some standards exist for a specific company. Other standards for graphics are general and used by most. Some of these are:

### Practical Design of Logic with HMI

From early in the chapter, A-B's design allows for a Push Buttons with the selection Multistate. The memory circuit can be turned on or off from the multistate button. The better memory circuit can also be turned on or off from the program. The memory circuit then must be able to report to the HMI the present state. The program for this function is left as an exercise.



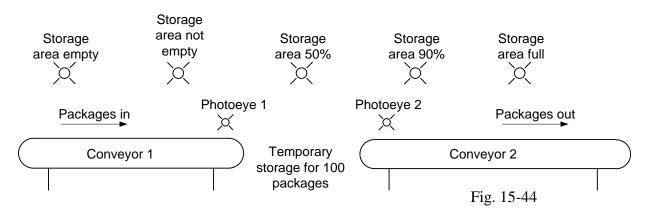
The use of multistate buttons to provide this logic is useful but not necessary. The use of a single button on the screen is an advantage in that one button can be used to turn on the memory, turn off the memory and display the present state of the memory. All three functions can be accomplished with a single button. All the time in advertising, we hear the ad for buy one, get one free or the 'two-fer' ad. This is a real 'three-fer'. Buy one button and get the 'start' button, the 'stop' button and the 'indicator' light in the same button.



#### Where to Put the Logic

The following conveyor system has five outputs, lights for percent complete of packages going down conveyor 1 to conveyor 2. Write a program to turn on these lights based on the fact that packages must pass photo-eye 1 to enter the storage area and pass photo-eye 2 to exit. This program was solved in Ch. 8 using greater/less-than statements and discrete outputs. It is possible now to turn on these outputs in the HMI program with no statements in the Ladder other than the Up-Down counter.

You may find it easier or better to provide the logic in the HMI or Ladder. There is usually a preference in most companies for one or the other or you may decide for them.



Consider Multi-State Indicators for the application above using RSView Studio as an example:

Multistate Ind	licato	r Properti	es			×		
General St	ates	Common	Connections					
Name			Tag / E:	xpression	Tag	Exprn		Eig 15 45
Indicator	+				***	•••		Fig. 15-45
							Ш	

	Ŧ
If Logical Relational Arithmetic Bitwise Functions Iags Check Syntax	

# Example: Expressions that return numeric values

For these examples, assume tag1 = 5 and tag2 = 25.

Expression	Returned Value
tag1	5
<b>tag1 + tag2</b> (arithmetic operator)	30
~tag1 (bitwise operator)	-6
SQRT(tag2) (mathematical function)	_

# Example: Expressions that return true/false values

Expression	Returned value
<b>tag1 &gt; 20</b> (relational operator)	1 (true) if tag1 is greater than 20
	0 (false) if tag1 is less than or equal to 20
Industry\Valve AND Municipal\Valve	1 (true) if both valves are open
(logical operator)	0 (false) if one or both valves are closed

### Example: Controlling visibility with If-Then-Else logic

To create a graphic object that is to be visible only when tag1 exceeds a specified value

- 1. Draw the object.
- 2. In the Visibility animation dialog type the expression:

If (tag1 > 55) Then 1 Else 0

3. Specify that the object is to be visible when the expression is true.

### From 1973, an HMI That Worked After Months of Development

In 1973, I was intensely involved in a project involving an HMI for three stations for a glass manufacturing line. The program was written entirely in assembler on a 16 k machine (IBM System 7). The HMI portion was to be six screens for the entire line including the three stations. What happened to that project was the complete explosion of need for a manual back-up that could over-ride the automatic system originally envisioned. The only response was to create a manual back-up system that expanded the original six screens to 66 screens. Along the way, a complete diagnostic system was also developed since electricians and operators could not be expected to keep the line operational without good data concerning the sensors on the machine and which if any were not operating correctly. The eventual HMI program consisted of about 23k assembly statements found in 66 individual over-lay programs running in a 384 word over-lay area of the computer.

The program was a tree-designed system with operators being asked a series of choices from which an operator successively moved further out a branch of the tree. In the end it worked and was accepted by the operators and plant engineers. It took way too long and enveloped my time for more than a year. Hopefully the lessons learned from experiences such as these would be helpful for future engineering efforts.

One of the two learned was the conflict between production and the hypothetical original design that insisted on complete automatic control. The manufacturing environment was not ready for the complete automation of the theorist. If a production supervisor found it more advantageous to have the destination of glass be on a take-off point on the right side instead of the left, he wanted that flexibility. The automatic system did not give that kind of flexibility and he refused to use the system until developed to his needs. The other lesson was that of complete diagnostic reporting. The operation was not going to succeed until every switch was operational (for glass tracking) and these switches were constantly falling into disrepair due to the moving glass. It was only after a dedicated line on each display showed the active status of any broken switch and its exact location was the tracking system allowed to function.

#### From 1998, an HMI System Divided

In 1998, I worked on a system for a steel company that saw the dividing of the project into multiple assignments with four engineers working simultaneously on two PLC/HMI programs. The PLCs were Modicon and the HMI was Wonderware. In this project I was responsible for the PLC program for process control. My counterpart worked on the HMI portion from India. Two individuals programmed the steel movement portion of the project from their office in Toronto, Canada. The most difficult portion was the HMI portion for the steel movement. It

involved a number of rules for the handling of the steel. These rules were total written from the Wonderware software platform.

The rules for the handling of the steel were similar to those of a baker with an oven big enough to handle multiple cakes with various baking times and temperatures. If a cake was in the back of the oven and was ready to come out, could a new cake be inserted in its place given the time over-lap of a second or third cake presently in the oven or ready also to be placed in the oven? The rules allowed for proper bake time and temperature for each individual cake while controlling the temperature of the overall oven.

From an article in the dmcinfo website, we see an intriguing topic:

#### Multiplexing Arrays of UDTs in TIA Portal V14 By Jason Mayes Dated 12/06/2016

### It's back!

When V13 SP1 was first released, one of the features I was most excited about was the ability to multiplex arrays of custom UDTs on a PLC using a simple index tag on an HMI. This feature allowed me to use a single faceplate or screen to display or modify data from multiple objects by simply changing a single tag. I was so excited, I even <u>wrote a blog post</u> about it then! Unfortunately, it was a feature to be short lived and, due to some unexpected consequences, mysteriously disappeared with the release of Update 3.

Now that <u>TIA Portal V14</u> has been released, the very first thing I did was look to see if it was back. And I'm in luck!"

You may ask what is an UDT and how does a UDT save time with graphic development. This is a good article to start exploring how this is done.

Also, we are introduced in the above article to an open library for Siemens. It is described in the following:

"The Siemens Open Library was developed by <u>DMC</u>, <u>Inc.</u> over several years. Through a joint collaboration with Siemens Industry, DMC documented the library and released it as an Open Source Library open for anyone to use and distribute. The Library is Open Source and will allow for users to contribute content to help enhance the overall availability of functionality to the greater Siemens User Community."

Again this sounds like something to look into in your job if Siemens is the plc of choice.

### Summary

This chapter is the only chapter solely dedicated to the HMI graphic panel. In Ch. 7, we had a short tutorial involving getting the Siemens' HMI attached to the S7-1200. That was an introduction to the HMI panel and useful for encouraging students to use the panels instead of wiring to buttons and lights. This chapter expands on that first experience in that both Siemens and A-B are discussed as well as types of product.

The basic panels for both manufactures are introduced and explored. Buttons as well as other devices are built. Some examples of how to use various graphics are included as well. The chapter ends with a discussion of graphics standards and a common problem that I commonly refer as the 'three-fer' button. The chapter is not meant to be an exhaustive study of HMI panels but as a starting point for students needing to learn some graphics before launching their careers.

While this chapter begins the broad development of HMI panels, the design of panel interfaces and screen interfaces continue in subsequent chapters, especially the chapter on motion and the chapter on pid control.

### Lab 15.1 Revisit - The Cash Register

The basic lab is copied from Chapter 7 as follows:

Design a simple cash register similar to one found at McDonald's or Burger King. To do this, determine a menu of five or six items from the restaurant. Also, include a Total button or a clear button or possibly both. Also, include a means for backing out of a mistake without starting over from zero. Display the cost of the total order in the PLC at an address in the data table. Use floating point math and you are encouraged to do so.

For example:

Whopper Combo	Whopper	Cancel Last	
Whopper Dbl Combo	Fries	New Order	Fig. 15-
Whopper Jr Combo	Drink	Total/Tax/ Optional	

Find the approximate prices from a McDonald's or Burger King for the items chosen. When an item is entered, its count is incremented automatically by one. If a button is entered multiple times, the count is incremented to display the total count. If a mistake is made, the attendant must be able to back up at least one entry and erase the last item or decrement that item by one.

Hints to the base lab:

Notice that counters may be referenced as either Count Up or Count Down. If the count is counting up, the count is incremented in rung 0000. If the count is counted down, the count is decremented in rung 0001. Individual inputs are used to increment each product choice. However, to decrement the count, a separate button labeled "Cancel Last" is used. This button must remember the last product chosen and decrement that item. Use the logic in chapter 7 "Relay Instructions" to remember when a button was pushed.

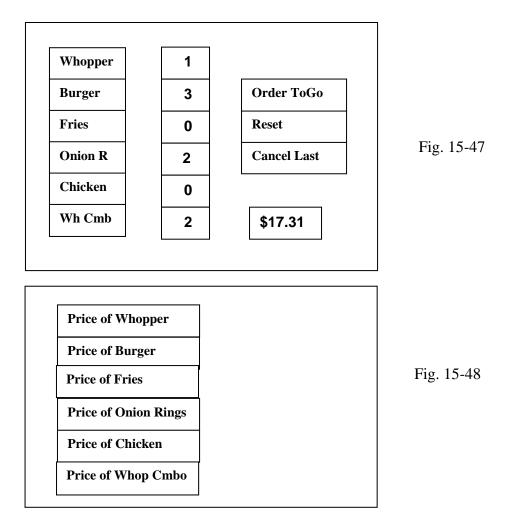
Use the Count Up/Count Down logic for holding active counts for the various items in the cash register.

Make the following changes for the application:

1. Display the total price for the order on the screen. Use Floating Point numbers where possible. Display totals in \$xx.xx format.

46

- 2. Add a second screen to allow the manager to change base prices for each item. Do not include a password to move from screen to screen.
- 3. Include a button to add 6.25% tax if not "To Go" for the order.
- 4. Include a 'live' count of the number of each item ordered.
- 5. Create means for going from Screen 1 to Screen 2.
- 6. Screens should resemble the following for Lab 15.1:



**Lab 15.2** Build the Conveyor application as described above for Siemens. Then build the same application for A-B. Compare the two. You will need to write PLC logic to move the elements and increment counts. You do not need to copy the programs included but may write your own programs.

Lab 15.3 Build the graphic for one of the graphics represented in Fig. 15-24 thru Fig. 24 - 39

**Lab 15.4** Build the graphic and logic for the graphics represented in Fig. 15-43 using a Siemens PLC.

**Lab 15.4** Build the logic and graphic for Fig. 24 – 44.

### Questions

- 1. How would you build the following pushbuttons in Siemens? Describe:
  - Momentary Maintained Latched Multistate Interlocked Ramp
- 2. Describe how to demonstrate the flow of a liquid of red color through a series of pipes. Several hand valves are in the path of the flow. How would one describe this graphic with A-B, Siemens?
- 3. You are assigned the task of describing a ride at an amusement park graphically. The ride is a zip-line. You may place sensors at the beginning and end of the ride. Describe the graphic using A-B and then Siemens controllers and HMI.
- 4. For the following device, what is the animation property used to move the clock's hand.

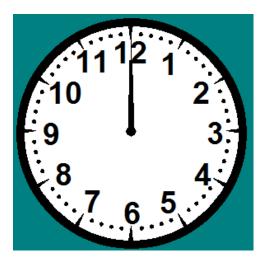


Fig. 15-49

- 5. Why would one design a PLC and HMI system with OPC and Visual Basic rather than one of the packages described in this chapter?
- 6. Describe how the programmer would animate the bottle on the conveyor moving right to left for Allen-Bradley, for Siemens.
- 7. Describe how A-B and Siemens shows each bottle being entered into the case of bottles.

- 8. Using either A-B or Siemens ladder logic, write a cash register program that uses an accompanying HMI program with only three items [hamburger, fries, drink]. Include logic to calculate the total price ignoring tax. Ignore the 'cancel last' button and combo logic as well.
- 9. Using A-B or Siemens ladder logic, write a program that could be used to show the movement of a bottle across a conveyor belt and populate a case. Use a case of 12. Use a start and stop button to start the operation. Use a reset button to set the bottle counter in the case back to 0 and allow another operation to fill the case with bottles.
- 10. If you were moving a program from real pushbuttons to an HMI, what one thing must you do?

# Setting the PG/PC Interface for Runtime Simulation

In order to establish a connection between runtime simulation at the PG/PC and theS7-1200 CPU, first we have to set the PG/PC interface to TCP/IP.

1	
	Open the system control • with "Start > System control" (start menu for the simplified access to the programs under Windows XP) • or with "Start > Settings > System control" (for the classical start menu as in earlier Windows versions).
2	Now double click on the icon "Set PG/PC interface"
3	In the tab "Access Path", set the following parameters: 1. For the access point of the application, select from the drop down menu "S7ONLINE [STEP 7]". 2. In the list of Interface Parameter Assignment Used, highlight the interface "TCP/IP(Auto) -> with your network card that is connected directly to the panel and the control system; for example, Intel(R) PRO/100 VE. 3. Then click OK and confirm the message that follows with OK also. Set PG/PC Interface Access Path LLDP Access Path LLDP Access Point of the Application: [STONLINE [STEP 7] -> TCP/IP -> Inte(R) PRO/100 VE Ne [Standard for STEP 7] Interface Parameter Assignment Used: TCP/IP -> Intel(R) PRO/100 VE Ne (Ac Properties Delete (Assigning Parameters to Your NDIS CPs with TCP/IP Protocol (PFC-1006)]

#### Starting the Configuration in Runtime

Click on the button Start runtime.



Adjus	t your computer's settings						View by: Large icons *
pr	Action Center	色	Administrative Tools		AutoPlay	*	Backup and Restore
<b>N</b>	Color Management		Communication Settings		Credential Manager	P	Date and Time
•	Default Programs	G.	Desktop Gadgets	-	Device Manager	-	Devices and Printers
	Display	C	Ease of Access Center	P	Folder Options	A	Fonts
	Getting Started	<b>4</b> 3	HomeGroup	æ	Indexing Options	•	Intel® Rapid Storage Technology
6	Internet Options	5	Java		Keyboard		Location and Other Sensors
1	Memory Card Parameter Assignmen (32-bit)	ð	Mouse	群	Network and Sharing Center		Notification Area Icons
	NVIDIA nView Desktop Manager		Performance Information and Tools	-	Personalization	0	Phone and Modem
8	Power Options	風	Programs and Features	1	Recovery	8	Region and Language
-	RemoteApp and Desktop Connections	<b>a</b> ,	Set PG/PC Interface (32-bit)	0	Sound	ą	Speech Recognition
۲	Sync Center	1	System		Taskbar and Start Menu		Troubleshooting
1	User Accounts		Windows CardSpace	鼬	Windows Defender	-	Windows Firewall
æ	Windows Update						

Alternately, the following must be set up to get the PID program to run properly with the HMI simulate mode:

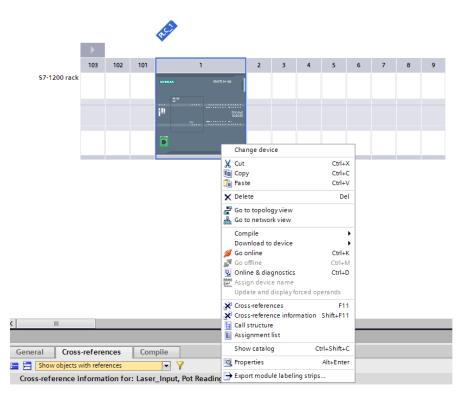
Click on the SetPG/PC Interface box above:

We are going to use the Siemens program TIA Portal V14 in order to run the program given for the ball-in-tube lab. This program will be used to download the PLC program but not the HMI program.

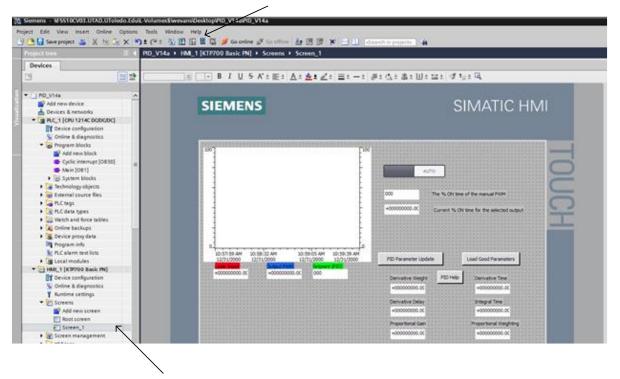
Choose the third of the Broadcom choices. Click OK.

p	Action Center	8	Administrative Tools	A	utoPlay	1	Backup and Restore
2	Color Management		Communication Settin	igs 🧃 C	redential Manager	r	Date and Time
•	Default Programs		Desktop Gadgets	-	evice Manager	100.00	Devices and Printers
4	Display	٢	Ease o Access Path U	e DP / DCP   PNIO Ada			Fonts
	Getting Started	•3	Homes Access Port of		Net/Oneme 57kx Gigabit Contr 💌	P	Intel® Rapid Storage Technology
	Internet Options	1		ter Assignment Used		3	Location and Other Sensors
1	Memory Card Parameter Assignmen (32-bit)	ð	Mouse Stones	berne 57kx Gigabit Con	. Degrostics		Notification Area Icons
•	NVIDIA nView Desktop Manager		Perfor Broadcom M	ietXtreme 57xx Gigabit ietXtreme 57xx Gigabit ietXtreme 57xx Gigabit	Cooy .	3	Phone and Modem
1	Power Options	â	Progra (Parameter assig	ment for the IE-PG acco s with TCP/IP Protocol		P	Region and Language
-	RemoteApp and Desktop Connections		Set PG (32-bit			₿	Speech Recognition
۲	Sync Center	1	System				Troubleshooting
82	User Accounts	1	Windo	-	CancelHelp	4	Windows Firewall
	Windows Update				,		

This allows the Siemens program to run the HMI program in simulate mode. Then download the program to the PLC. Do not download the HMI program since we do not have the HMI to download to.



Then click on the HMI's Screen, Screen\_1. Notice the Start Simulation button turn blue. It now allows the student to run the HMI via simulation mode from the screen of the pc.



We need to find the variables that need to be written to the historical data. Click on Laser Input and get the tag Laser\_Percent. This tag is one to be written to the historical data logger.

	<ul> <li>PLC_1 [CPU 1214C DQ/DQ/DC]</li> <li>HMI_1 [KTP700 Basic PN]</li> <li>Device configuration</li> <li>Online &amp; diagnostics</li> <li>Runtime settings</li> <li>Screens</li> <li>Add new screen</li> <li>Root screen</li> <li>Screen_1</li> <li>Screen_1a</li> <li>Screen management</li> <li>HMI tags</li> <li>Connections</li> <li>HMI laarms</li> <li>Recipes</li> <li>Historical data</li> </ul>		0-10:57:55 12/31/2 000000	000 12	58:32 AM /31/2000 +000000		10:59:39 AM 12/31/2000
~	5 Scheduled tasks Details view	~	VO field_1 [VO field]		- Europte	Terda	
	Name		Properties     Anir       General     Appearance       Characteristics     Lougust	G	Events	Texts	cent

Next, go to the Historical data tag under HMI tags. Set the variables from the HMI screen above that are to be saved. Fill in the appropriate fields and start the historical logger. This circular file will contain the data from the analog data saved. Then run the program and run the Historical data logger. It is a circular file and will wrap around after the table fills up.

		Data logs								
🖫 Online & diagnostics	^	Name 🔺	Storage location	Data records	. Path	Logging method	Number o			
🍸 Runtime settings		📘 Data_log_1	TXT file (Unicode) 🔽	500 🗘	۱USB_X60.1۱	🖌 Circular log 🛛 🔽	10			
🔻 📄 Screens		<add new=""></add>								
📑 Add new screen										
Root screen										
🚬 Screen_1							>			
🕨 📷 Screen management	=	<								
🕨 🔙 HMI tags		Logging tags			-					
迄 Connections		Name 🔺	Process tag	Acq	uisition mode Logging cycle	High limit	Low			
🖂 HMI alarms		kanc 🛓	Laser_Percent				Ø-			
📑 Recipes		<add new=""></add>	Laber_reident	Oyo			2.			
🛄 Historical data		4 Idd Hens								
5 Scheduled tasks										
🔛 Text and graphic lists										
Lear administration										