

CorsairHMI Corrections

12/4/2018

Manuals in this series

[CHMI_Overview.pdf](#)

[CHMI_Operator.pdf](#)

[CHMI_Developer.pdf](#)

[CHMI_Designer.pdf](#)

[CHMI_Experts.pdf](#)

CHMI_Corrections.pdf – This Manual

[CHMI_BatteryMon.pdf](#)

[CHMI_Glossary.pdf](#)

Contents

Introduction	3
Scramble Code Entry.....	4
Hook Codes	4
Video Switching.....	5
Digifort Driver	5
Digifort Quick Start	5
Digifort Driver Documentation	8
Pelco Driver	18
Vicon	22
Intercom.....	22
The Harding DXL Intercom Driver	22
The Harding DXI Intercom Driver	25
The Telecor T3 Intercom Driver	26
Duress	32
The Centurion Driver.....	32
Guard Tour	32
Access Control.....	32
Honeywell Pro-Watch	33
Keyscan	41
Doors.....	41

Introduction

The CorsairHMI program comes with a license file that determines what features of the program are available to an end user. Corrections features are things that are used in computer systems for jails and prisons. This manual is used to describe those features. Some of them are available in any version of the Corsair program. Some are available only if the customer has purchased a Corrections version license.

Scramble Code Entry

CorsairHMI includes a special feature for entreating operator logon PIN numbers in corrections systems. It is a numeric entry keypad that appears on a touchscreen. The positions of the numbers on the keypad are different every time it appears. They can be made to change each time a digit is entered. This is helpful in direct supervision situations where inmates may see the touchscreen. They cannot learn the operator code by memorizing the positions of the touches. The developer must carefully select size, color, and font options for the code entry window to make it impossible for the inmates to read. The selections should be checked after the installation is complete. The code numbers should be changed periodically.

Hook Codes

During interface operation the operator may 'hook' to different placements on the screen by touching the surface of a touch monitor or by clicking the mouse. Many types of placements can be hooked without being operated. When they are hooked the status window displays data pertaining to the placement.

Corrections applications frequently require that the PLC has some knowledge of what placement the operator is hooked to. An example is when door control, closed circuit TV, and intercom are all integrated. When the operator touches the placement corresponding to a cell door the video switcher should show him a view of the door and the intercom should be ready to talk to the speaker inside that cell. This is accomplished through the use of one or more hook code values that can be associated with each screen placement.

The computer database has a record for each interface computer mode that is running the application. It provides fields that may be used to link each computer up to four hook code devices. These devices are used to specify what PLC data addresses receive the hook code values when the operator hooks a placement.

For example, an integer device tagged 'Camera #' could be placed on a PLC. Another named 'Intercom Station' could be placed on the same or a different PLC. The first hook code device for a computer would then be set to 'Camera #' and the second to 'Intercom Station.' Camera numbers for a video switcher could range from 1 to 16. Intercom station numbers could be 3-digit values from 100 to 999. As each hookable placement is put on the screen it would have a camera number entered into the first hook code value and an intercom station number entered into the second hook code value. The PLC would then control the video switcher and the intercom so that they would track with the actions of the operator.

The PLC-based hook code approach to integration tends to work more efficiently than direct computer control of video and intercom when there are multiple computer interface nodes. Typically, each interface computer uses different addresses for its hook code devices. They may be created as separate tags. A common alternative is to utilize computer indexing on the hook code devices. This permits using

the same device tag for hook codes on all computers in the database as long as each computer has a unique addressing index.

The computer writes the hook value(s) into the appropriate devices one time when the placement is hooked and the hook code values are non-zero. It writes zeros one time when the placement is unhooked. Changing directly from one placement to another that has non-zero hook codes will cause the new values to be written without a zero value between them.

Video Switching

A Corsair operator in a corrections facility typically has a computer monitor that displays CorsairHMI screens. He may have one or more separate monitors that show him closed-circuit TV (CCTV) images from within the building. Some systems may display as many as 16 images on sections of a single monitor. These images are 'static' because the same camera is displayed at the same place all the time. Some systems may have only a single CCTV 'event' monitor. An event monitor changes which camera it is displaying based upon an event. The event may be the operator touching an icon on the Corsair touchscreen or someone pushing an intercom call button. Many systems have a combination of static and event-driven CCTV images.

CCTV systems are of two types. One type is based upon conventional video signals that travel through a shielded coaxial cable. The other type uses 'IP' cameras that feed into an Ethernet network.

Digifort Driver

Digifort Quick Start

This procedure allows a developer to complete initial testing of the communication between the CorsairHMI program and a Digifort VMS server.

Install the Corsair program on your computer

Create a c:\corsair folder on your computer.

Copy the corsair.exe program into it.

Right-click on the program and send it to the desktop as a created shortcut.

The Corsair icon should now show on your desktop.

Right-click on the Corsair icon and pick properties.

Set the icon to 'run as administrator'. This may be under the 'Advanced' options.

Go to the www.corsairhmi.com website.

Under the Public Downloads find the 'free 4-hour corsair runtime license'

Download the 'CHMI_Demo4.cky' license file.

Place the license file in your C:\corsair folder. This license will allow the Corsair interface to run for 4 hours. After that the interface will shut off with no loss of data. It can then be restarted for another 4 hours. There is no limit to the total runtime with this license in 4-hour segments.

Run the Corsair program.

Click on 'Users/Change Levels'. Type 'admin' into the password.

Select the Developer – Administrator option. Click on OK.

The lower-right status bar should now include 'Dev: Admin'.

Pick the 'Setup/Computer Properties' menu option.

On the Startup Tab pick Development-Administrator.

On the Security Tab enter a base session name of 'Corsair'.

On the Security Tab check 'Allow Exit', 'Allow Minimize', and 'Prevent Running a Second Copy'.

Click on OK to accept the Computer Properties.

Click on the 'Setup/Save Properties' menu option.

Initialize an Empty Corsair Model

Pick the 'Setup/Model List' menu option. Pick 'Edit/Empty' to create an empty model.

From the main menu pick 'File/Save File' and OK to save the model data.

Determine the IP address of your Digifort Server

Corsair requires a 4-byte IP address for the server. If you are accessing a server over the Internet and you only have a domain name an address will have to be determined. Pick the 'Tools/TCP Expert' menu option. Click on 'Get IP from Name'. Type the domain name in the 'Entered Name' field. Click on the 'Get' button. Write down any discovered 4-part IP address. A 6-part address will not work with Corsair.

If the address is not a static IP it may change at some time based upon the server's Internet Service Provider. The address on the Corsair data source will have to be changed whenever this happens. Permanent installations must have static IPs.

Determine the required Digifort information

You will need the Digifort User name and password that you will use. This information will come from the administrator of your Digifort server.

You will need to have a Surveillance Client running on a computer. It does not have to be the same computer that is running the Corsair software. You will need to know the name of the client's monitor on the Digifort virtual matrix.

Generate an initial Database

Click on the 'Edit/Database/Generation' menu option.

Do not use the 'Empty' button. Use the 'Clear' button to clear out any generator data.

Click on 'Sessions'. Click on '1'. Enter 'Corsair' for the ID of the first session (or computer). OK twice to get back to the main database generator window. A check should appear next to the Sessions button showing that it will generate a session.

Click on 'Screens'. Click on '1 Ovr' to specify an overview screen. Enter 'Digifort Screen' for the ID of the screen. OK twice to get back to the main database generator window. A check should appear next to the Screens button showing that it will generate a screen.

Click on 'Drivers'. Click on 'D1'. Enter 'Digifort Driver' for the ID of the driver. Pick 'DIGIFORT – Digifort VMS' for the type. Click on 'Accept' and not on OK.

Click on 'S1' for Source 1. Enter 'Digifort Monitor' for the ID of the data source.

Check the 'Create Reserved Addresses' checkbox. Enter the IP address of the Digifort server.

Click on Oks and Close until you get to the main database generator window. A check should now appear by the 'Drivers' button to show that it will generate a driver.

Click on the 'Generate' key and answer the 'OK' prompt to do the generation.

Close the Generation window. Corsair will warn you that you are about to lose unsaved generation data. Click on OK to go ahead and lose the data.

From the main menu pick 'File/Save File' and OK to save the model data.

Additional Special Setup data entry

Pick the 'Edit/Data/Drivers' option on the main menu. Arrow up until the 'Digifort Driver' cell is highlighted. Press 'Z' to zoom.

Enter the master user name and password. Arrow to the desired selection and press F2 to edit.

Escape from the Special Setup window. Escape from the driver database.

Pick the 'Edit/Data/Sources' option on the main menu. Arrow up and to the right until the 'Node' field is highlighted. Press F2 to edit it. Type in a value of '1' and press enter for Node 1.

Arrow left until the 'Digifort Monitor' cell is highlighted. Press 'Z' to zoom.

Arrow through the fields and press F2 to edit each one. The port number can be left at zero. The Monitor ID needs to be set to the Surveillance client's monitor name on the virtual matrix. The same user name and password need to be entered here.

Escape from the Special Setup window. Escape from the Source database.

From the main menu pick 'File/Save File' and OK to save the model data.

Use the Register Monitor

Click on the 'Interface' checkbox to start run-mode operation.

Click on 'Tools/Data Source/Registers' to open the Digifort Monitor window.

Use the 'Get' buttons to get cameras, monitors, styles, views, users, and Global Events.

Use the 'Number' button to number the cameras with a first value of 1.

If there are Global Events click on the 'G Events' button to view them. Use the F2 key to enter nonzero Corsair numbers on each global event.

From the main menu pick 'File/Save File' and OK to save the model data.

The following documentation gives more detail as to how to operate the Monitor window.

Digifort Driver Documentation

The Digifort driver enables the Corsair program to switch IP camera images on a monitor that is acting as a Digifort surveillance client.

An example prison system would have 3 Digifort server computers each recording up to 200 cameras. These servers are known as East, West, and Central. Their IP addresses on the Corsair network are 1.1.1.1, 1.1.1.2, and 1.1.1.3. The system is designed with no more than 200 cameras per server. Each camera needs a unique Corsair ID number. The developer has decided to use the numbers 1 through 200 for cameras on the East server, 201 through 400 for cameras on the West server, and 401 through 600 for cameras on the Central server.

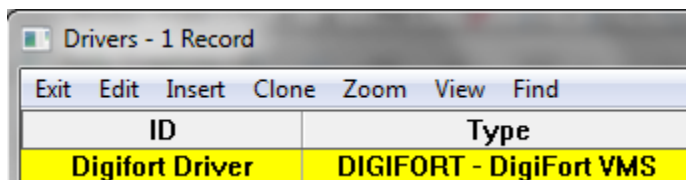
Besides the 600 cameras and 3 servers the network includes 6 operator workstations. Each workstation consists of two computers. One computer is running Digifort's surveillance client. The other computer of each pair is running the CorsairHMI program. These 12 computers use IP addresses from 1.1.1.4 to 1.1.1.15. The six workstations are identified by their user. They are known as Bob, Carol, Ted, Alice, Sally, and Roger.

The Digifort developer must enter 200 cameras into each of the 3 servers. He assigns them 600 unique names. He sets up user names and passwords for each of the 6 users. He sets up permissions for each user. Bob, Ted, and Alice can view the east sallyport camera but Carol, Sally, and Roger cannot. Sally is the only one that can see the booking camera.

The Digifort developer must create a seventh user name and password for a 'master' user. This user must be able to see any camera that can be viewed by any of the six users. The Master user login is used by the Corsair program to get the list of cameras from each server.

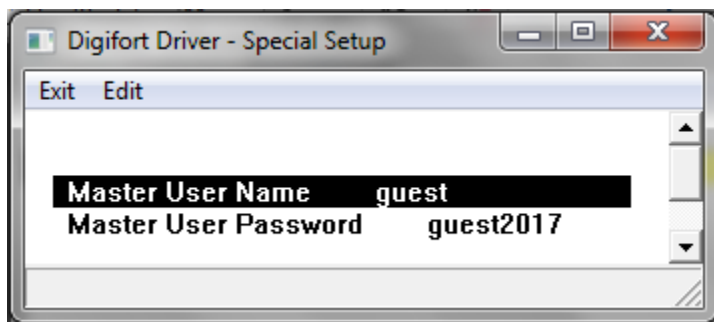
The Corsair developer intends for each of the 6 workstations to be using identical Corsair model files. The Corsair authority system will be used to determine the capabilities of each computer. He creates 6 session records named Bob through Roger. Each session name must have an index value entered for use with session-indexed tags. Index numbers from 0 to 5 are entered.

The developer then creates a driver record with the Digifort driver type.



ID	Type
Digifort Driver	DIGIFORT - DigiFort VMS

There are no IP addresses on the driver record. Zooming on this record allows him to enter the Master user name and password.



Master User Name	guest
Master User Password	guest2017

The next step is to create Data Source records under the driver. A data source record can represent a Digifort server, or a surveillance client monitor, or both. This system with 3 servers and 6 monitors will require 6 data source records. It could also be done with 9 data sources – 3 that represent servers and 6 that represent monitors.

Digifort Driver - Data Sources - Local - 6 Records				
Exit Edit Insert Clone Zoom View Find				
ID	IP	Node	Real	Driver
East Server, Bob Monitor	1.1.1.1	0	Yes	Digifort Dr..
West Server, Carol Monitor	1.1.1.2	1	Yes	Digifort Dr..
Central Server, Ted Monitor	1.1.1.3	2	Yes	Digifort Dr..
Alice Monitor	0.0.0.0	3	Yes	Digifort Dr..
Sally Monitor	0.0.0.0	4	Yes	Digifort Dr..
Roger Monitor	0.0.0.0	5	Yes	Digifort Dr..

Each data source is set to be 'Real' or 'Live'. A data source record that represents a server needs the IP address of that server. A data source that represents a monitor but not a server gets a zero IP address.

Each record has a node value that matches the index of the corresponding session. Other drivers do not use the node number in this way. When a tag on this driver is 'Session-indexed' the index number on the session record corresponds to the node number on a data source. That data source describes a monitor on the virtual matrix that is used by that session.

This configuration assumes only 1 Digifort 'Monitor' per Surveillance client workstation. If more monitors are created on the virtual matrix additional data source records will have to be developed. There will still be only three sources for the servers.

The next step for the developer is to zoom on each data source record.

East Server, Bob Monitor - Special Setup	
Exit Edit	
TCP Port Number (0 defaults to 8601)	0
Monitor ID	Bobs Monitor
Viewer User Name	Bob
Viewer User Password	Bobs Password
Replace same Style	No
Replace same Camera	No

The port number is the port that Corsair will use to connect to the server. If it is left at zero the default value of 8601 will be used. The name of the monitor must be entered. Monitor names must be unique throughout the whole virtual matrix system. There cannot be a 'Monitor 1' on Bob's client and also on Carol's client. The monitor's Viewing user name and password are entered here. If a workstation surveillance client has 2 Digifort monitors they will have different ID names but the same user name and password.

The Replace options are normally set to 'No'. This will cause Corsair to use the 'DoNotReloadSameStyle=TRUE' and 'DoNotReplaceSameObj=TRUE' parameters when loading a screen style and showing an object. Setting them to 'Yes' eliminates these parameters. Consult the Digifort

documentation for information about how they work. These parameters are used on data sources that are servers. They have no effect on data sources that only represent monitors.

The next step is to load the camera list for each of the 3 server data sources. These are the 3 records that have nonzero IP addresses. It is done from the Digifort driver's register monitor window.

The Camera 'Get' key loads the camera database from the server using the Master User name and password. This list is opened with the 'cameras' button.

East Server, Bob Monitor - Cameras					
Exit View					
Number	Name	Description	Type	Ad...	Model
1	01	Client parking	1 - IP Camera		Axis P3367-V
2	02	People Counter	1 - IP Camera		Axis M3114-R
3	04	Entrance micro camera	1 - IP Camera		Axis P8514
4	14	Sala Suporte 360Â° fisheye	1 - IP Camera		Samsung SNF-7010V
5	15	Digifort InSight **	0 - Unknown ??		Digifort InSight
0	33	Training Room 2 - DOME PT	1 - IP Camera		Axis M5014
0	35	Training Room 4 - model CAM 1	1 - IP Camera		Axis M1114
0	36	Training Room 5 - model CAM 2	1 - IP Camera		Axis Q1614
0	37	Training Room 6 - model CAM 3	1 - IP Camera		Axis P1346

The database contains the Corsair number for the camera, a camera name and a description. The type of the camera record is shown along with a Model designation. Some Master users are allowed to see an address for the camera.

The type field show which records represent IP cameras. The developer may wish to delete other types of records by blanking out their fields. The remaining cameras need to be numbered. This can be done automatically using the register monitor windows 'number' button or it can be done manually by entering a nonzero number into each record. Corsair cannot switch to a camera that has a zero number. The developer may wish to use the database 'sort' option when he is done numbering the cameras.

Cameras are only listed under server data sources with nonzero IP addresses. The camera databases under data sources with zero IPs are not used. Each camera that is listed under a driver must have a unique nonzero Corsair number. This is enforced by the decision to use the numbers 1 through 200 for cameras on the East server, 201 through 400 for cameras on the West server, and 401 through 600 for cameras on the Central server.

The Monitor 'Get' key loads the active monitors database from the server using the Master User name and password. This list is opened with the 'Monitors' button.

East Server, Bob Monitor - Active Monitors					
Exit View					
Monitor	User	Address	Object	Type	Description
100	videowall	10.1.10.11		0 - No object	
2	videowall	10.1.10.11	06	1 - Live camera	RecepÃ§Ã£o
3	videowall	10.1.10.11	29	1 - Live camera	Dome Subsolo
4	videowall	10.1.10.11	29	1 - Live camera	Dome Subsolo
5	videowall	10.1.10.11	27	1 - Live camera	Sala IntegraÃ§Ã£o

This database shows the active virtual matrix monitors at the time the 'Get' button is pressed. This data can be constantly changing. The developer must clear and then get the database again to get a current value. The Active Monitors database is meant as a diagnostic aid for developers.

The developer must be sure to save the Corsair model file anytime changes are made to the cameras and monitors databases.

The 'Display' buttons on the Digifort Monitor are used to test camera switching to different spots on the monitor.

The 'Styles' button is used to view a list of screenstyle numbers. The 'Get' button reads style numbers from a server. Each nonzero number represents a different pattern of monitor spots on a screen. The 'Set' button can be used to send an empty style to a monitor.

The 'Views' database lists public screen views that are available to all surveillance clients. It does not list screen views that are limited to a user. Each view has a name and a screen style number. Corsair can send a view to a monitor if the developer gives it a nonzero Corsair number. Corsair numbers must be unique across servers. The same number cannot be used for a view on more than one server.

The 'Users' database lists user names that have been entered into the Digifort server. The Master and Viewer user names that are entered under the driver and data source record must match entries from this list.

The 'Global Events' database lists global events that have been configured in the server. These are not manual events. They are global events that the Master user has the right to access. Each event has a name and a description. Corsair can trigger an event if the developer gives it a nonzero Corsair number. These numbers must be unique across servers.

The 'Manual Events' database lists manual events that have been assigned to cameras in the server. Each event has a Corsair camera number, a name, and a description. Corsair can trigger the event if the developer gives it a nonzero Corsair number. These numbers must be unique across servers and not collide with any of the numbers that have been assigned to global events.

Corsair numbers can be repeated across Cameras, Views, and Events. There can be a number twelve on each. There cannot be two Cameras that both have the number twelve.

The Event Tigger button is next to a number entry box. The developer can enter a Corsair number that matches either a Global or Manual event. When the Trigger Button is pressed the event is triggered.

There are several Reserved Tag addresses for use with the Digifort driver. They can be created using the 'Res Addr' button when doing single-record editing of the data source record.

Tag	Type	Start	End	Format	Chnge	Size	SI
Camera Addresses	String	Camera A...	Camera A...	78	Yes	20	No
Camera Descriptions	String	Camera D...	Camera D...	78	Yes	20	No
Camera Models	String	Camera M...	Camera M...	78	Yes	20	No
Camera Names	String	Camera N...	Camera N...	78	Yes	20	No
Camera Numbers	Integer	Camera N...	Camera N...	U5.0	Yes	20	No
Camera Types	String	Camera T...	Camera T...	78	Yes	20	No
Monitor Client Addresses	String	Monitor Cl...	Monitor Cl...	78	Yes	10	No
Monitor Client Users	String	Monitor Cl...	Monitor Cl...	78	Yes	10	No
Monitor Names	String	Monitor N...	Monitor N...	78	Yes	10	No
Monitor Object Descriptions	String	Monitor O...	Monitor O...	78	Yes	10	No
Monitor Object Names	String	Monitor O...	Monitor O...	78	Yes	10	No
Monitor Object Types	String	Monitor O...	Monitor O...	78	Yes	10	No
Spot 0 Camera	Integer	Spot 0 Ca...		-5.0	Yes	1	Yes
Spot 1 Camera	Integer	Spot 1 Ca...		-5.0	Yes	1	Yes
Spot 2 Camera	Integer	Spot 2 Ca...		-5.0	Yes	1	Yes
Spot 3 Camera	Integer	Spot 3 Ca...		-5.0	Yes	1	Yes
Spot 4 Camera	Integer	Spot 4 Ca...		-5.0	Yes	1	Yes
Spot 5 Camera	Integer	Spot 5 Ca...		-5.0	Yes	1	Yes
Spot 6 Camera	Integer	Spot 6 Ca...		-5.0	Yes	1	Yes
Spot 7 Camera	Integer	Spot 7 Ca...		-5.0	Yes	1	Yes
Spot 8 Camera	Integer	Spot 8 Ca...		-5.0	Yes	1	Yes
Spot 9 Camera	Integer	Spot 9 Ca...		-5.0	Yes	1	Yes
Spot 10 Camera	Integer	Spot 10 C...		-5.0	Yes	1	Yes
Spot 11 Camera	Integer	Spot 11 C...		-5.0	Yes	1	Yes
Spot 12 Camera	Integer	Spot 12 C...		-5.0	Yes	1	Yes
Spot 13 Camera	Integer	Spot 13 C...		-5.0	Yes	1	Yes
Spot 14 Camera	Integer	Spot 14 C...		-5.0	Yes	1	Yes
Spot 15 Camera	Integer	Spot 15 C...		-5.0	Yes	1	Yes
Spot 16 Camera	Integer	Spot 16 C...		-5.0	Yes	1	Yes

Some of the reserved addresses for tags with the Digifort driver are designed to not use session-indexed ('SI') addressing. These tags may be indexed as conventional arrays with a size that is greater than 1. Each index of these tags corresponds to a monitor. If a value is written to index 3 of the tag it applies to the monitor on the data source with a node number that is set to 3.

Some of the reserved addresses for tags with the Digifort driver are designed to use session-indexed ('SI') addressing. These addresses have a 'SI' suffix. The array size field is left at 1. When a value is written to one of these tags Corsair looks at the address index value of the computer's session record. It finds the data source whose node address matches that value. The value is used for that monitor. Typically the session-indexed version of the address is what is used for hook codes.

User Login IP Security

The Digifort administrator may want to use User Login IP security to guarantee that users only use the correct surveillance clients. In our example system there must be two IP addresses entered for each server – the address of the user's surveillance client and the address of the user's Corsair computer.

Surveillance Client Configuration

The Surveillance client will need to have at least one monitor configured for use on the Virtual Matrix. This monitor gets a name that will be used for the 'Monitor ID' entry under the data source record. There are two check boxes in the Virtual Matrix settings:

Show object origin information

Blink border when an object is loaded on the monitor

Both of these checkboxes should not be checked.

Screen Style Tags

The Digifort system comes with some standard screenstyle layouts. These divide a computer monitor into a number of sections called 'Spots'. Spots are numbered starting at one. The Digifort administrator can create custom screenstyles. Each standard or custom screenstyle has a unique nonzero identification number. It is shown on the right side of the monitor when the administrator is modifying the style.

There are two tags that are used to set screenstyles into a monitor. They are 'Set Screenstyle' and 'Set Screenstyle SI'. Both are double integers. They are written with nonzero screenstyle identification numbers.

The 'Set Screenstyle' address can be an indexed address. If it has a size of 1 the style is applied to the monitor corresponding to the data source that the tag is on. If the address has a size of more than one which index is written determines what monitor is used. If a style identification number is written to index 3 of the tag it applies to the monitor on the data source with a node number that is set to 3.

The 'Set Screenstyle SI' address is session-indexed with a size of 1. When a value is written to this tag Corsair looks at the address index value of the computer's session record. It finds the data source whose node address matches that value. The style is sent to that monitor.

View Tags

The Digifort system can save public screen views. Corsair can display these screen views on a monitor. This is done with two tags. They are 'Set View' and 'Set View SI'. Both are integers. They are written with nonzero Corsair numbers that match entries in the Views database.

The 'Set View' address can be an indexed address. If it has a size of 1 the view is applied to the monitor corresponding to the data source that the tag is on. If the address has a size of more than one which index is written determines what monitor is used. If a number is written to index 4 of the tag it applies to the monitor on the data source with a node number that is set to 4.

The 'Set View SI' address is session-indexed with a size of 1. When a value is written to this tag Corsair looks at the address index value of the computer's session record. It finds the data source whose node address matches that value. The view is sent to that monitor.

Corsair uses the ?? user name and password to set screen views.

Spot Camera Tags

The 'Spot # Camera' tags are integer tags that can be used to control what camera appears in each spot of a screenstyle. Spot 00 is used for a full-screen view. Spots 01 through 16 are for individual spots on a screen style (layout). The addresses are available in both normal and SI 'Session-Indexed' versions.

Writing these tags with zero has no effect. If Corsair writes the value 6 into an index of the 'Spot 02 Camera' tag it will look through each server's camera database. It will search for a record with a number of 6 and a name that is not blank. Once it finds a camera it must find a monitor. This is determined differently for a normal tag versus a session-indexed tag. If it is a normal tag the index number that is written with the value is used to determine the monitor. If it is session-indexed the addressing index of the computers base session is used to determine the monitor. If a monitor is available the computer issues an HTTP ShowObject command to the Digifort server that contains the camera. This command uses the spot number that is indicated by the tag address.

Corsair uses the View user name and password to set a camera on a spot.

Compound-Indexed Spot Camera Tag

The regular Spot # Camera tags do not allow dynamic indexing across spot numbers. There is a special address that allows this capability. This 'Any Spot Camera' address has unique operating rules. It is session-indexed with an array size that is greater than 1.

To be continued . . .

Starting and Stopping Recording

Corsair can start and stop recording on individual cameras. This requires some changes in the Recording Settings on the server. Instead of 'Continuous Recording' the 'Recording by Schedule' option needs to be selected. The Recording Scheduling button leads to the Scheduling window. The 'Record by Event' action is used when Corsair is to start and stop the recording.

The 'Start Camera Recording' tag address is a double integer that is written with a nonzero camera number to start recording of that camera. It is not indexed so it has a size of 1.

The 'End Camera Recording' tag address is a double integer that is written with a nonzero camera number to stop recording of that camera. It is not indexed so it has a size of 1.

The 'Camera Recording Switches' tag is an array of switches with one index for each camera. When index 6 is turned on recording starts on camera number 6. When index 12 is turned off recording stops on camera number 12. The tag should have an array size one higher than the highest camera number.

Corsair uses the Master user name and password to start and stop camera recording.

Camera Privacy

The 'Set Camera Privacy' tag address is a double integer that is written with a nonzero camera number to activate privacy on that camera. It is not indexed so it has a size of 1.

The 'Reset Camera Privacy' tag address is a double integer that is written with a nonzero camera number to release privacy on that camera. It is not indexed so it has a size of 1.

The 'Camera Privacy Switches' tag is an array of switches with one index for each camera. When index 6 is turned on privacy is activated on camera number 6. When index 12 is turned off privacy is deactivated on camera number 12. The tag should have an array size one higher than the highest camera number.

Corsair uses the Master user name and password to switch camera privacy on and off.

Camera Privacy Groups

Corsair supports the development of privacy groups. A privacy group is a list of cameras whose privacy can be activated or released from a single tag. Each privacy group gets a nonzero number to identify the group. Each group can contain a large number of cameras.

The 'Set Group Privacy' tag address is an integer that is written with a privacy group number to activate privacy for the cameras on that group. It is not indexed so it has a size of 1.

The 'Reset Group Privacy' tag address is an integer that is written with a privacy group number to release privacy for the cameras on that group. It is not indexed so it has a size of 1.

The 'Group Privacy Switches' tag is an array of switches with one index for each group. When index 6 is turned on privacy is activated for the cameras of group number 6. When index 12 is turned off privacy is deactivated for the cameras of group number 12. The tag should have an array size one higher than the highest privacy group number.

Corsair uses the Master user name and password to switch camera privacy groups on and off.

Alerts

Corsair can trigger global or manual events on a Digifort server. One way to do this is by writing a nonzero Corsair number value to the 'Trigger Event' tag. The program first looks at the Global Events database for a matching number. If it does not find it then it goes to the Manual Events database. When it finds the number if the Master User has the right the event is triggered.

A common way to trigger Digifort events is using the Corsair 'Trigger Alerts' block. The block is documented in the Corsair Designer manual. The driver includes two addresses that are used as parameters on the block and a third that it uses for itself.

The 'Alert Active Switches' tag is used for the Result parameter of the 'Trigger Alerts' block.

The 'Alert Done Switches' tag is used for Parameter F of the 'Trigger Alerts' block.

The block's Parameter C gets a memory tag or a tag from another driver.

The developer enters in Corsair numbers for the events associated with each element of these tags. This is done from the 'Alerts' button on the register monitor.

Corsair uses the Master user name and password to trigger Digifort events.

Multiple Camera Switches

Corsair can be used to simultaneously switch camera views on multiple spots of multiple monitors from a single trigger source. This may be called 'salvo switching'. A database can be set up to describe thousands of Camera/Monitor/Spot combinations from hundreds of triggers.

To be continued . . .

Pelco Driver

The CorsairHMI program can be hooked to a Pelco video switcher. An operator's station at a prison can have a video monitor next to the Corsair computer screen. As the operator selects doors on the touchscreen the Pelco equipment can switch the monitor to views from the camera closest to each door.

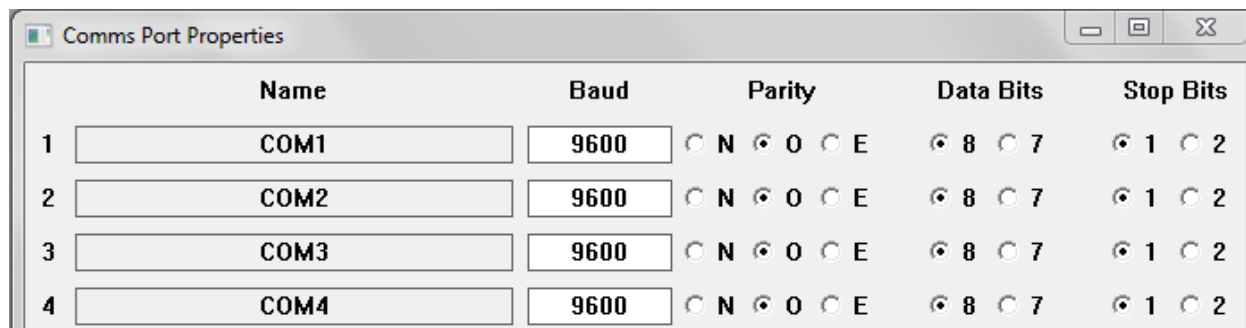
Driver Capabilities

As of this writing the driver can be used to switch what camera is displayed on a monitor. It can be used to start a sequence in the forward direction, start a sequence in the reverse direction, or hold a sequence. It can be used to match the switcher's date and time clock settings to the Corsair computer. With some systems it may be used to set a Camera Title.

The driver currently does not have functions to control camera pan, tilt, zoom, focus, or iris.

Comms Port Configuration

The connection between the Corsair computer and the switcher is typically RS-232. The Corsair driver that controls the equipment uses Pelco's ASCII protocol. The ASCII protocol uses 1 start bit, 8 data bits, odd parity, and 1 stop bit. The baud rate is usually 9600.

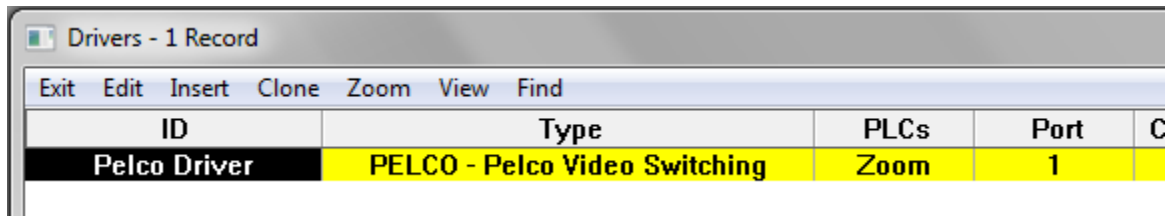


The screenshot shows a window titled 'Comms Port Properties' with a table of four COM ports. Each row represents a port (COM1 to COM4) and its configuration. The columns are Name, Baud, Parity, Data Bits, and Stop Bits. The Baud rate is set to 9600 for all ports. The Parity is set to Odd (O) for all ports. The Data Bits are set to 8 for all ports. The Stop Bits are set to 1 for all ports.

	Name	Baud	Parity	Data Bits	Stop Bits
1	COM1	9600	<input type="radio"/> N <input checked="" type="radio"/> O <input type="radio"/> E	<input checked="" type="radio"/> 8 <input type="radio"/> 7	<input checked="" type="radio"/> 1 <input type="radio"/> 2
2	COM2	9600	<input type="radio"/> N <input checked="" type="radio"/> O <input type="radio"/> E	<input checked="" type="radio"/> 8 <input type="radio"/> 7	<input checked="" type="radio"/> 1 <input type="radio"/> 2
3	COM3	9600	<input type="radio"/> N <input checked="" type="radio"/> O <input type="radio"/> E	<input checked="" type="radio"/> 8 <input type="radio"/> 7	<input checked="" type="radio"/> 1 <input type="radio"/> 2
4	COM4	9600	<input type="radio"/> N <input checked="" type="radio"/> O <input type="radio"/> E	<input checked="" type="radio"/> 8 <input type="radio"/> 7	<input checked="" type="radio"/> 1 <input type="radio"/> 2

Driver Configuration

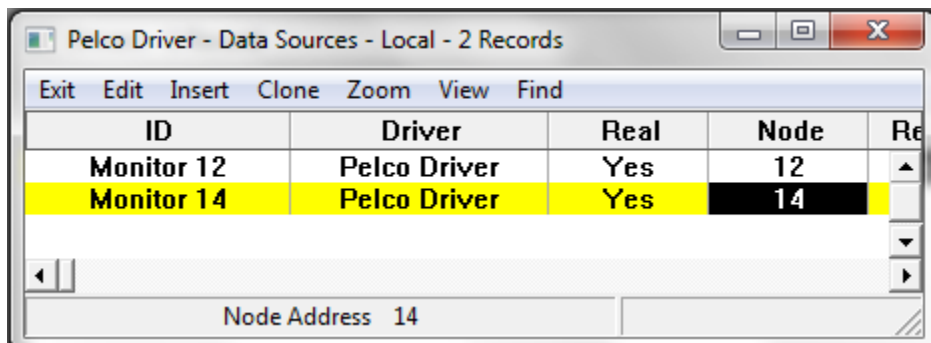
The Corsair program needs to have the development level set to 'Admin' so that development work can start. A driver record must be created. It gets the 'PELCO – Pelco Video Switching' type. The port number on the driver must be set to the desired serial port. Most commonly it is one. The ID can be any desired name.



The screenshot shows a window titled "Drivers - 1 Record" with a menu bar (Exit, Edit, Insert, Clone, Zoom, View, Find) and a table with the following data:

ID	Type	PLCs	Port	C
Pelco Driver	PELCO - Pelco Video Switching	Zoom	1	

The next step is to zoom on the drivers Data Source ('PLC') zoom field to enter one or more data source records.



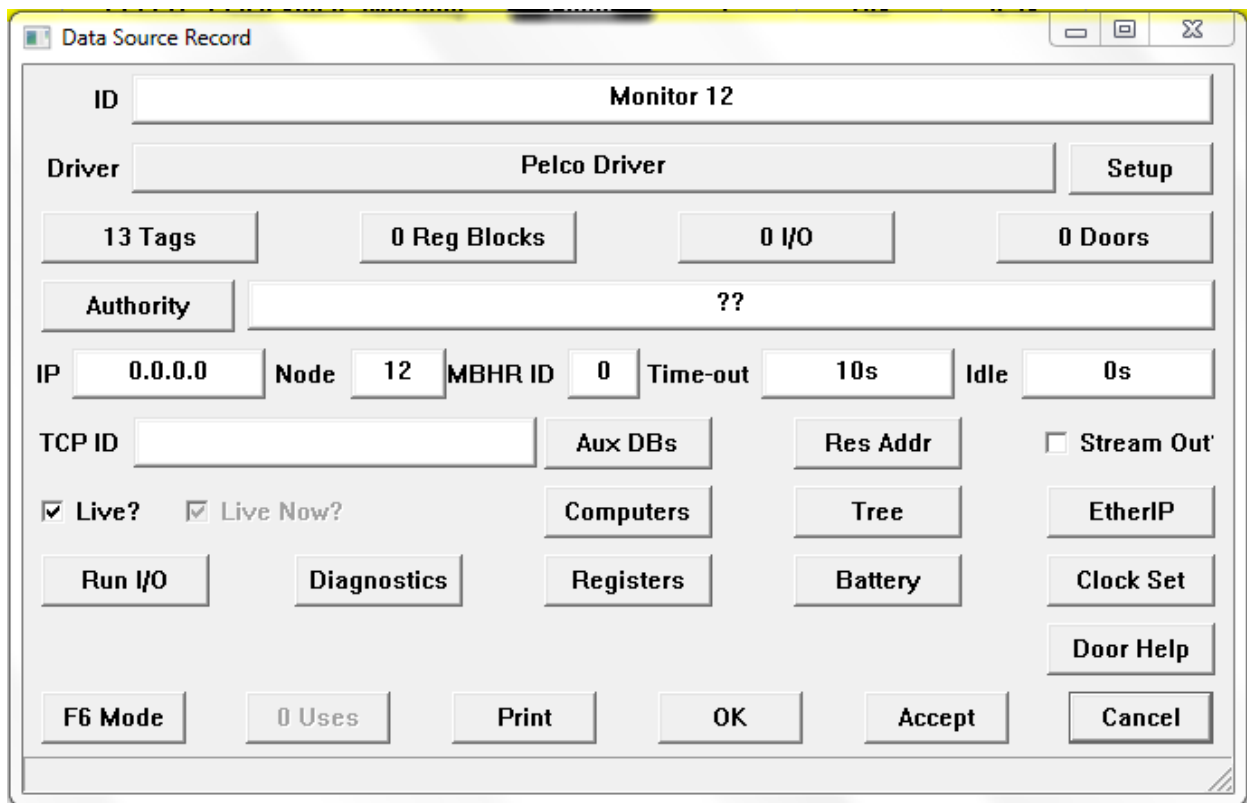
The screenshot shows a window titled "Pelco Driver - Data Sources - Local - 2 Records" with a menu bar (Exit, Edit, Insert, Clone, Zoom, View, Find) and a table with the following data:

ID	Driver	Real	Node	Re
Monitor 12	Pelco Driver	Yes	12	
Monitor 14	Pelco Driver	Yes	14	

Below the table, there is a field labeled "Node Address" with the value "14".

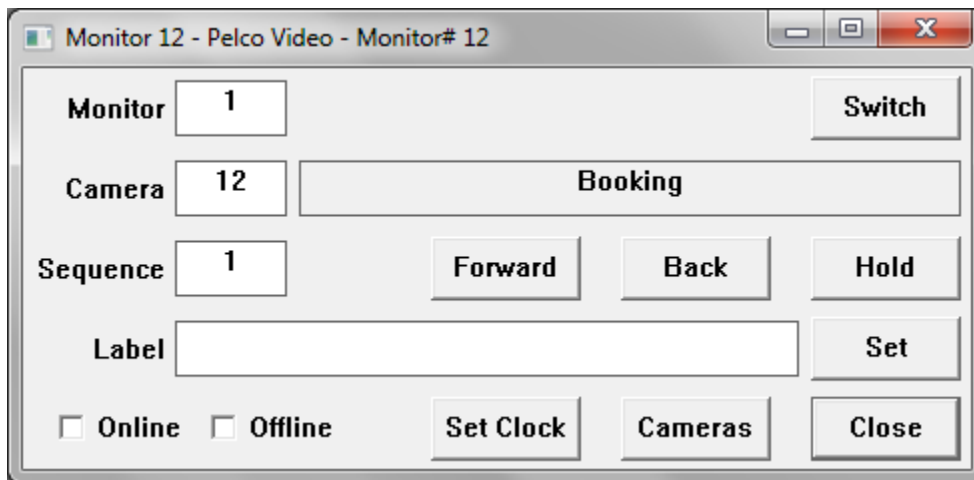
Each data source corresponds to a monitor. Anything can be entered for the ID name. The Node number of the data source corresponds to the Pelco monitor number. The Real flag must be set to 'Yes'.

Pressing F6 on each data source record opens data source single-record editing. The 'Res Addr' button is used to create tags on the data source with addresses that will work for the driver.



The 'Data Source Record' window is a configuration interface for a monitor. It features a title bar with standard window controls. The main area contains several input fields and buttons. At the top, the 'ID' field is set to 'Monitor 12'. Below it, the 'Driver' is set to 'Pelco Driver' with a 'Setup' button to its right. A row of four buttons shows '13 Tags', '0 Reg Blocks', '0 I/O', and '0 Doors'. The 'Authority' field is set to '??'. The 'IP' field is '0.0.0.0', 'Node' is '12', 'MBHR ID' is '0', 'Time-out' is '10s', and 'Idle' is '0s'. Below these are 'TCP ID', 'Aux DBs', 'Res Addr', and a 'Stream Out' checkbox. A row of checkboxes includes 'Live?' (checked), 'Live Now?' (checked), and buttons for 'Computers', 'Tree', and 'EtherIP'. Another row contains 'Run I/O', 'Diagnostics', 'Registers', 'Battery', and 'Clock Set'. At the bottom right is a 'Door Help' button. The bottom row includes 'F6 Mode', '0 Uses', 'Print', 'OK', 'Accept', and 'Cancel' buttons.

Corsair has a Data Source Register Monitor window that is designed for use with this driver.



The 'Monitor 12 - Pelco Video - Monitor# 12' window is a control interface for a camera switcher. It has a title bar with standard window controls. The main area includes a 'Monitor' field set to '1' with a 'Switch' button. Below this, the 'Camera' field is set to '12' next to a 'Booking' field. A 'Sequence' field is set to '1' with 'Forward', 'Back', and 'Hold' buttons. A 'Label' field is empty with a 'Set' button. At the bottom, there are 'Online' and 'Offline' checkboxes, and 'Set Clock', 'Cameras', and 'Close' buttons.

Camera switching is accomplished by entering a monitor number and camera number into the edit controls and then clicking on the 'Switch' button. A sequence is controlled by entering the monitor and sequence number and then using the 'Forward', 'Back', and 'Hold' buttons. A camera title is entered by entering the camera number and label and then clicking on 'Set'. The 'Set Clock' button is used to match the switcher's time to the Corsair computer.

The 'Cameras' button is used to access the Cameras database.

Camera #	Name
1	Front Entrance
3	North Sallyport Door
27	South Sallyport Door
12	Booking
0	
0	
0	

This database can optionally be used to enter numbers and names of cameras.

Each of the tags created by the Corsair program has a different function with the Pelco driver.

Tag	Type	Start	End	Format	Chnge	Size
Camera Code	Integer	Camera Code	Camera C...	U5.0	Yes	2
Cameras	String	Cameras	Cameras[9]	78	Yes	10
Clock Set	Switch	Clock Set			Yes	1
Comment A	String	Comment A	Comment ...	78	Yes	10
Comment B	String	Comment B	Comment ...	78	Yes	10
Label Camera	Integer	Label Camera		U5.0	Yes	1
Label String	String	Label String		78	Yes	1
Numbers	Double Int	Numbers	Numbers[9]	U6.0	Yes	10
Seq Backward	Integer	Seq Backward	Seq Back...	U5.0	Yes	2
Seq Forward	Integer	Seq Forward	Seq Forwa...	U5.0	Yes	2
Seq Hold	Integer	Seq Hold	Seq Hold[1]	U5.0	Yes	2

Camera Code is an arrayed tag with a size of 2. A single-element write of a camera number to index 0 of this tag causes the switcher to place that camera on the monitor that has been entered as the data source node number. This tag is commonly used as a Hook Code tag. A Corsair script can be used with a data move block to put values into both elements of the Camera Code tag. In that case, the zero (first) index is the camera number and the one (second) index is the monitor number. In this way any camera can be directed to any monitor. The Seq Forward and Backward tags work the same way where the first element is the sequence number and the optional second element is the monitor number. Writing any value to the Seq Hold tag stops any sequence on the data source's monitor. If both elements of the Seq Hold tag are written the second element is the monitor number.

Turning on the Clock Set tag triggers a download of the time from the Corsair computer to the Pelco switcher.

Vicon

Corsair contains a driver to switch cameras on monitors using a Vicon video switcher.

Intercom

The Harding DXL Intercom Driver

Corsair can control a Harding MicroComm DXL intercom using an Ethernet connection.

Harding DXL Configuration

The Harding programmer must define a Host Port on an Exchange in a DXL system. The Host Port must be set to be Ethernet and not Serial. It needs an IP address that is compatible with the network that the Corsair computer is on. The TCP/UDP port number should be left at the default value of 10000 for the first host port on the Exchange. If there are multiple Exchanges each one gets a unique IP address but each one can use the same port number of 10000. If more than 20 computers talk to a single exchange more host ports will have to be created. The next host port should get an address of 10001, then 10002, and so on.

The next step is to define the host port protocol. The protocol to use is 'ASCII Messages', not 'Register Based Messages'.

The Messages tab of the Harding software has some checkboxes for options for the messages. One option is for status messages to 'Use Response Message Format'. It should not be checked. Response messages should have the 'Respond to All Host Commands' option checked. The 'Use Status Message Format' option should be checked. Under Acknowledgments the 'Wait for Acknowledge Messages' option should not be checked. These settings are different than the Harding software's default settings.

The Monitor tab of the Harding software has a 'Monitor Connection for Faults' checkbox. It should be checked. The Timeout should be set for the default 10 seconds.

Corsair Development for the DXL

The Corsair program needs to have the development level set to 'Admin' so that development work can start. A driver record must be created. It gets the 'DXL - Harding Instruments DXL Intercom' type. The ID can be any desired name. The PLC network IP address on the driver must be set to match the DXL intercom Exchange.

The next step is to zoom on the drivers Data Source ('PLC') zoom field to enter one or more data source records.

Each data source corresponds to a master. Anything can be entered for the ID name. The Node number of the data source must be used to enter the master's nonzero dial number. Master dial numbers greater than 65535 will not work here. The Real flag must be set to 'Yes'.

Most systems will require session-indexed tagging to be used for some Harding tags. With these systems it is not necessary to create a separate data source for each master station. Create just one source and give it the dial number of the first master in the system.

At this point Register Monitoring for the data source may be used to check communications to the intercom.

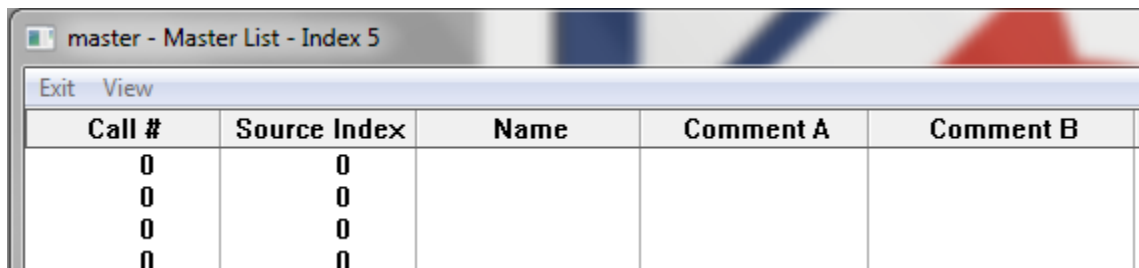
If no tags have been created the master number on the title bar is the dial number of the master that the window is controlling. It can be used to originate calls to a station, to another master, or to initiate a page.

The next step is to create tags on the data source record. Select 'Edit' 'Data' 'Sources' and select the intercom data source. Pressing F6 opens data source single-record editing. The 'Res Addr' button is used to create tags on the data source with addresses that will work for the driver. The sizes of some of these tags will have to be adjusted as explained later.

After tags have been created the 'List' and 'Groups' buttons on the Register Monitor are used to open the auxiliary databases that are used by the driver. Remember that any changes to an aux database are saved as part of the .cap Model file. There will be no warning prompt if the developer exists without saving aux database data.

The developer must define a unique index number for each master station, intercom station, station group, and paging zone that is used in the system. Master stations get the first index numbers starting at 1. They correspond to session index values in the Sessions database.

The first database to work with is the master list.



Call #	Source Index	Name	Comment A	Comment B
0	0			
0	0			
0	0			
0	0			

The master list corresponds Corsair sessions to master station dial numbers. Any computer that uses the DXL driver has a base session name. This is the name that is set using the Security tab of the Computer Properties window. The index of that session in the sessions database must be a nonzero value. The first column of the master list is the call (dial) number of the master. The second column is the corresponding Corsair session index number. Both columns must be nonzero for the row to be valid. A name for the master should be entered. The optional comments are for the architecture printout.

Assume that the sessions database has three records:

Central Control - session index 1

North Control - session index 2

South Control – session index 3

Assume that the master list has three records:

Call #101 - Session Index 1 – Name 'Central Control Intercom Master'

Call #201 – Session Index 2 – Name 'North Control Intercom Master'

Call #301 – Session Index 3 – Name 'South Control Intercom Master'

The Central Control computer will be using the intercom master with dial number 101. The North Control computer will be using the intercom master with dial number 201. The South Control computer will be using the intercom master with dial number 301.

The next aux database to work with is the Intercom Station List.

Call #	Tag Index	Name	Comment A	Comment B
0	0			
0	0			
0	0			
0	0			
0	0			
0	0			
0	0			
0	0			
0	0			
0	0			

The Call # is the dial number for each station. The Tag Index is a nonzero number that locates the intercom station in an array of tag data. These tag indexes normally start after the range of values that is used for the indexes in the Master list. They can be entered into the database in any order. The same tag index should not be used for more than one intercom station. The same tag index should not be used for both a station and a master. Both the Call # and Index columns must be nonzero for the row to be valid. A name for the station should be entered. The optional comments are for the architecture printout.

The 'Master Names', 'Master Index', 'Master Numbers', 'Master Comment A, and 'Master Comment B' arrays should all be sized slightly larger than the maximum count of masters in the system. These tags are never session-indexed.

The 'Station Names', 'Station Index', 'Station Numbers', 'Station Comment A', and 'Station Comment B' arrays should all be sized slightly larger than the maximum count of stations in the system. These tags are never session indexed.

The 'Enable Switch', 'End Call', and 'Next' tags should all be left with a size of 1. Usually session indexing will be set to 'Yes' for these tags. If the interface turns on the button with the 'Next' address Corsair looks to see if the tag is session indexed. If it is not session indexed Corsair uses the dial number that is stored in as the node address of the data source. If it is session indexed Corsair finds the dial number for its session in the Master List aux database. In either case, if the dial number is zero the 'Next' command is not sent to the DXL.

The Harding DXI Intercom Driver

Corsair can control a Harding MicroComm DXI intercom using an Ethernet connection.

The Telecor T3 Intercom Driver

Initial Development

Initial Testing – the T3 Register Monitor

Required Documents

There are two documents besides this one that the Telecor application developer needs. The other two are printed from the Corsair program. They are the About Driver printout for the T3 driver and the Corsair Telecor T3 protocol document.

The About the Driver Printout

The driver About printout can be printed without printing all the other drivers. Go through Edit/Data/Drivers and arrow to the driver that is used to talk to the T3. Press F6 to go to driver single-record editing. The About button is used to make the printout. It lists all the possible addresses that can be used for tags with this driver. A data source must be created on the driver. Pressing F6 while on the data source record leads to single-record editing of the source. The 'reserved addresses' button is used to create the tags. Some tags will be created with an array size of 1. These tags will usually be left at that size. Other tags that are initially created with a size of 10 will need to have their size adjusted manually by the developer. For best performance these tags should not be sized much larger than what is required.

The end of the About printout shows the value enumerations that are used for the 'Station LED State', 'Station Page State', and 'Network LED State' tags.

Corsair Telecor T3 Protocol Printout

The second document that the developer should print is the Telecor T3 Protocol section of the Corsair Application Manual. This section details the messages that the Corsair program expects from the T3 and the commands that it will send to the T3. The person that does the T3 programming will need to study this so that he can determine what he needs to do.

The manual section is printed through File/Print/Application Manual. Click on 'Print' to select a printer. Click on 'Print Nothing' to shut off all manual sections. Go to the 'Reference' tab. Check 'Telecor T3 Protocol' and click on 'OK'.

The 'T3 Default Messages' are messages sent from the T3 to the Corsair computer. They correspond to the default messages that are suggested by Telecor. 'Default Commands' work in the opposite direction. They go from the Corsair computer to the T3.

Special Messages and Commands will have to be configured as a part of the T3 programming work.

Station State

The driver supports three tag addresses that are integer values corresponding to the status of intercom stations. They are 'Station LED State', 'Station Page State', and 'Network LED State'. Each of these tags should be sized to accommodate the number of indexes that are needed on the system.

The 'Station LED State' tag contains integer values ranging from 0 through 6 to indicate what the station is doing. The About Driver printout for the Telecor T3 driver gives the definitions for each of these states. The state value can be used to modify what the operator sees on the screen for each intercom station. The developer may choose to place an ellipse to simulate an LED. The state value can be used as an index into a color set for the ellipse so that the LED changes color. A more common application would be to use an icon set. The state value would be used to determine what icon the operator sees.

When the operator does paging the Telecor T3 does not send messages to the Corsair computer to show it what stations are involved. This would require too many messages for the serial port. The developer may want to show on the computer screen which stations are being paged. This is the function of the 'Station Page State' tag address. Normally the value of the page state is identical to the value of the LED state. When the station is being paged the page state value changes to a 7. This enables a separate color or icon in the set to show paging status.

If a system does not utilize paging either the 'Station LED State' or 'Station Page State' tags can be used. Performance would be improved by using 'Station LED State' and deleting 'Station Page State'. If it is desired to indicate paging the 'Station Page State' tag is the correct choice. For this tag to work properly entries must be made in the 'Page Zones' and 'Page Stations' auxiliary databases. The section on Paging Indication that is later in this document shows how that is to be done.

Network LED State

Multiple T3 intercoms can be linked together on a Telecor network where each T3 communicates to a separate Corsair computer over a serial port. Assume that the building A T3 has intercom stations 101 to 160 on it. The building B T3 has intercom stations 201 to 260 on it. It is possible for the Building B T3 to talk to station 122 in Building A using the Telecor network. When this happens the 'Station LED State' and 'Station Page State' tags in the Building A Corsair computer will not show the operator that someone else is talking to the station. This becomes possible with the 'Network LED State' tag.

The first step for this to work is that the T3 programmer must enable the option that sends all messages from all the T3s to every computer. The Corsair developer must then enter a separate data source into his application database for each T3 that is on the network. Contact your Corsair dealer for more information as to how to do this. The second document that the developer should print is the Telecor T3 Protocol section of the Corsair Application Manual. This section details the messages that the Corsair program expects from the T3 and the commands that it will send to the T3. The person that does the T3 programming will need to study this so that he can determine what he needs to do.

The manual section is printed through File/Print/Application Manual. Click on 'Print' to select a printer. Click on 'Print Nothing' to shut off all manual sections. Go to the 'Reference' tab. Check 'Telecor T3 Protocol' and click on 'OK'.

Dial Number Indexing

Each intercom station on the T3 system has a dial number. 41903 would be a possible number. There are several Tag data addresses on the Corsair driver that are arrays. Each element of these arrays corresponds to a station dial number. The 'Station LED State' tag is an example of one of these arrays. Corsair defaults to the simplest situation where array element indexes correspond directly to intercom dial numbers. Dial number 41903 corresponds to index 41903 of the array. This system is easy to develop because the tags for the icon for that station would use array index 41903.

The problem with simple dial number indexing is that the tags have to be sized for the highest dial number that is on the system. If dial number 41903 is present the tags must have a size of at least 41904 to allow for indexes 0 through 41903. This is true even if there are actually only 200 intercom stations on the system.

The Corsair program performs lots of processing on T3 tag data to calculate paging, call group, and Network LED status. With large tag array sizes this processing can become very slow and system performance will be unacceptable. The answer to this problem is to use the Stations auxiliary database. The Stations database is used to map Station dial numbers to Tag array indexes. Now our example tags can be sized at 200 instead of 41904. Dial number 41903 can be linked in the database to tag index 1. Tags on the icon for that station would now use an index value of 1 rather than 41903.

Synchronization

It is possible with systems where intercoms communicate with computers for the two processors to lose data synchronization. If this occurs the computer may not show a call-in that is present or it may show a call-in that is no longer active. Two synchronization techniques can be used with the Corsair T3 driver.

The first option for synchronization is for the Corsair computer to send a CLR clear command when it is first started. It then resets all of its data to zero to clear all calls. The T3 does the same thing. The assumption is that the computer and T3 are synchronized at this point and they will stay that way. The only advantage of this option is that it requires minimal programming on the T3. One of the disadvantages is that when a Corsair computer is offline and it restarts all T3 call status information is lost. Another disadvantage is that the system has no way to recover from lost communications characters.

The answer to these problems is the GCS Get Console State special command. The Corsair developer can block the sending of the CLR command on startup. He then programs a nonzero synchronization time interval on the data source record. Now the Corsair computer will send a GCS command when it starts and at regular time intervals after that. The T3 is to respond to the GCS with a complete listing of the status of all stations. The T3 programmer must configure the intercom to properly handle the GCS command.

Call Groups

Many times it is desirable to show an indication on the Corsair screen if one or more calls have been received from a group of stations. A graphic screen may show an overview of a 5-story building. There may be 5 'keys' (buttons) on the screen to jump to detail views of each of the 5 floors. The developer may want to have each key blink if there is an active call from that floor. He will want to assign 5 non-zero call group indexes. In this case indexes 1 through 5 will correspond to floors 1 through 5.

Three tag addresses are used to set up external page paths with the 'Call Groups' auxiliary database. The addresses are:

Group Index
Start Station Index
End Station Index

Each record of this database assigns a range of station index numbers to a call group. The database may look like this:

Group Index	Call Group Start	Call Group End
1	100	199
2	200	299
3	300	399
4	400	499
5	500	599
1	602	0
5	708	0

Zero is not a valid group index. A start or end value of zero is ignored. In this example a 3## call will indicate on the third floor call group. A 1## call or a call from 602 will show on the first floor call group. A 5## call or a call from 708 will show on the fifth floor call group.

The 'Call Group Active' tag address is used for the indicators that come on when the groups are active. The indexes of this tag correspond to the databases Group Indexes. Index 3 of this tag should be used to change color or blink the key for the third floor. Index 5 would be the indication for the fifth floor.

The 'Call Group Active' tag is calculated from the data in the 'Station Calls' tag. It will not work correctly if the 'Station Calls' tag is not present or is not large enough.

Paging Indication

Dial numbers can be configured by the T3 programmer to initiate pages to multiple intercom stations. If it is desired to have the 'Station Page State' tag properly show what stations are being paged the zone information must be programmed into the Corsair system. This is done with the 'Page Zones' and 'Page Stations' auxiliary databases.

External Pages

The Corsair software is capable of generating pages over the T3 system. Windows .wav wave audio files are the source of the sound. The output of the computer sound card is fed into the external page input of the T3. The application database contains data describing one or more talk paths for external pages. When the computer does a page it opens one of these talk paths, makes the page, and then closes the talk path. It can then open another talk path if desired. Only one external page talk path can be open at a time. The actual destination of an external page talk path is a property of the T3 programming. A path can go to only a few speakers or to an entire facility.

There does not have to be a unique external page talk path for each wave file that the computer uses. Several different wave files may use the same talk path. One of the first things that the developer must do is determine how many external page talk paths are required.

Five tag addresses are used to set up external page paths with the 'External Pages' auxiliary database. These tags must have a size at least equal to the number of external page talk paths (or larger). The addresses are:

Ext Page Index
Ext Page Name
Ext Page Source System
Ext Page Dest System
Ext Page Dest Station

The first thing that the developer must do is to define a list of index numbers starting at one for the external page paths. Assume that a building has 5 floors. The index 1-5 external page paths would be to page each of the individual floors. The index 6 path would be an all-page path for the entire building.

The Source System number on an external page is the system number of the T3 that the computer is connected to. If a nonzero system number is entered here the External Page Start command will use it. If this value is left at zero the computer will use the current value of the 'System Number' tag. If it is zero the computer will default to using a Source System value of one.

The required values for the Destination System entries are defined by the Telecor EPS command. The Destination Station values are the dial numbers of the pages. This example assumes 901 through 906.

Assuming a single T3 system the 'External Pages' database would look like this:

Index	Name	Source Sys	Dest Sys	Station
1	Floor 1	1	1	901
2	Floor 2	1	1	902
3	Floor 3	1	1	903
4	Floor 4	1	1	904
5	Floor 5	1	1	905
6	All Page	1	1	906

The indexes in this database correspond to elements in the 'Clear Talk Path', 'Request Talk Path', and 'Talk Path Ready' tags. The [0] element of each of these tags is not used. The [3] element of each of these tags is used to page the third floor. The [6] element of each of these tags is used to do an All-Page.

Three tag addresses are used to coordinate the programs sound logic with the Telecor T3 driver. Each of these tags must have a size at least equal to the number of external page talk paths plus one. They may be sized larger if desired. These addresses are:

Clear Talk Path
Request Talk Path
Talk Path Ready

The next development step is to define the desired sounds. The Sounds database is accessed under Edit/Graphics/Sounds. Each sound can consist of one or more wave files with an optional delay before each wave file is played. Zooming on the 'Files' zoom field allows listing the wave files. After the wave files are entered, escaping back to the Sounds database. The 'Play' field allows the developer to test for the expected sound.

The next step is to fill in the rest of the fields on the Sound record. The RTP Tag field is linked to the Telecor drivers 'Request Talk Path' tag. The RTP Index field is set between 1 and 6 to indicate the desired talk path – it would be 5 for a fifth floor page. The RTP Value field is always set to '1' for this driver. The TPR Tag field is linked to the Telecor drivers 'Talk Path Ready' tag. The TPR Index must be set to the same value as the RTP index. The CTP Tag field is linked to the Telecor driver's 'Clear Talk Path' tag. The CTP Index must be set to the same value as the RTP and TPR indexes. The CTP Value field is always set to '1' for this driver.

When the computer wants to use a talk path for an external page it sets a value of 1 into an index on the Request Talk Path tag. The driver looks at what index is nonzero to help it find the correct entry in the External Pages database. It uses this information to form and send an External Page Start command through the serial port to the Telecor T3. When the T3 replies that the command is successful the driver turns on the correct index of the Talk Path Ready tag. The Corsair computer plays the wave file. It then turns on the correct index of the Clear Talk Path tag. The driver sends External Page End to the T3. The sequence works like this:

Sound Program	Telecor Driver Program	Telecor T3
Set RTP tag to 1		
	Send EPS command	
		Open Page Talk Path
		Reply to EPS
	Set TPR tag to 1	
Play the sound		
Set CTP tag to 1		
	Send EPE command	
		Close Page Talk Path

The 'Timeout' field specifies how long the Corsair program will wait for the 'Talk Path Ready' signal from the Telecor. If it does not get TPR within this amount of time the wave file is played and the sequence continues normally. The timeout should be set to a value longer than the longest possible T3 response time. Note that External Pages should receive a very high priority in the T3 program for this system to work. If the timeout is set to zero or if a TPR tag is not listed the computer will play the wave file immediately without waiting for the T3 to open a path.

SQL Event Logging

Call-ins generated from the T3 may be logged into an SQL database using the Corsair event logging system.

Duress

Duress and man-down systems are used to alert others when an individual has been attacked or that some form of emergency help is needed. Corsair can interface to a wide variety of duress equipment.

The Centurion Driver

The Centurion driver is used to communicate with a Response Technologies Centurion Elite Duress system. This uses a serial connection. This driver is present in all versions of the Corsair program. A corrections license is not required.

The '??' tag address on a Centurion driver can be used as the parameter C 'Trigger Switches' input to a 'Trigger Alerts' block. The Active Switches Result of this block can be used to trigger Alerts on other drivers. This tag should only be used on one instance of the block. The '??' .. addresses perform the same function. They may be used for other instances of the Trigger Alerts block.

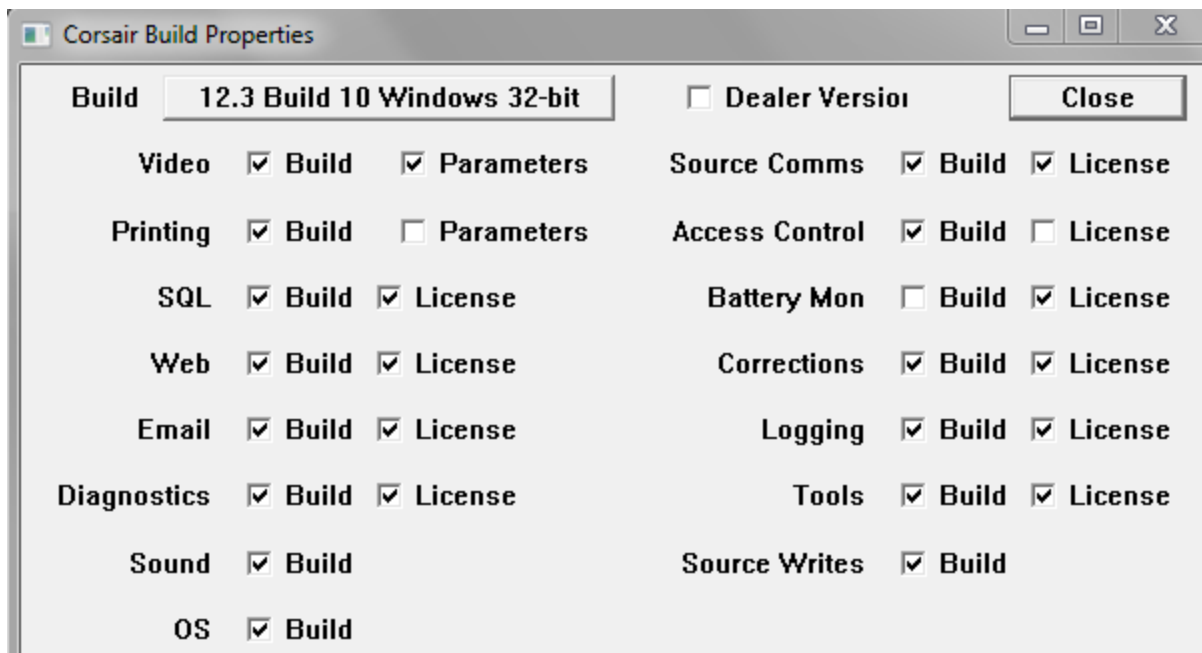
Guard Tour

Guard tour systems are used to verify that corrections officers make proper rounds inspecting parts of a facility.

Access Control

Access control systems are used to permit properly authorized individuals to get into restricted areas of a facility. Corsair can be used to interface between an access control system and a video management system. It can switch camera views on a monitor when a card is scanned. Some systems may show a picture of the individual that the card is issued to. If the person at the door matches the picture the operator opens the door.

The access control systems drivers in Corsair require both 'Access Control' and 'Corrections' capabilities. These may be verified from the main menu by clicking on 'Help'/'About' and selecting the 'Build' option. This opens the Build Properties window.



Both 'Build' and 'License' must be checked for Access Control and Corrections on this window. If 'Build' is not checked the developer must get a different version of the Corsair program. If 'License' is not checked CorsairHMI must be contacted for a different license file.

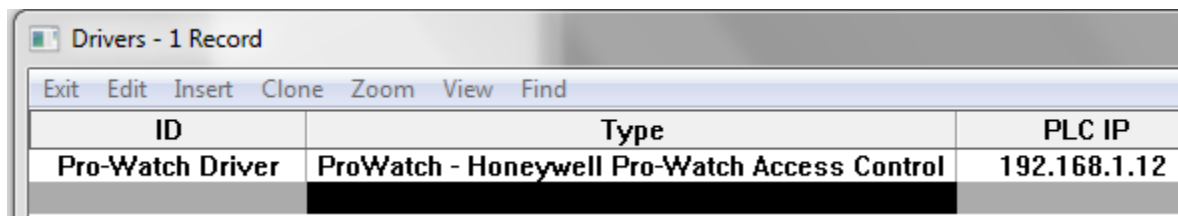
Honeywell Pro-Watch

The Corsair interface can be connected to a computer running the Honeywell Pro-Watch access control system. Corsair can lock and unlock doors and monitor alarms from Pro-Watch. Developing a system with this driver requires the user to have some knowledge of the Telnet feature of the Corsair's TCP Expert. He must also be able to use the JSON Expert. This is necessary because Corsair talks to Pro-Watch using the HTTP API. Pro-Watch doors and alarms have ID strings that are very long and difficult to enter. The work is simplified by having Corsair's Experts read in the data from the Pro-Watch database and save it in a file on the Corsair computer. This means that the access control system database must be completely developed before the Corsair database can be finished. The developer must set up Corsair to do the data read and then save the results in a JSON tree file. Corsair then uses that file to access the Pro-Watch program.

The first step with a Pro-Watch system is for the Pro-Watch developer to complete the Pro-Watch configuration database. This includes all doors and alarms. He must then configure the Pro-Watch API service. Typically, he will use TCP port 8734 for the REST access and port 8735 for the SignalR system. Corsair can use other port numbers if these defaults are not possible. The Pro-Watch computer must be installed on a network that is accessible from the Corsair computer. In this example Pro-Watch has a fixed IP address of 192.168.1.12. The Corsair computer has a fixed IP address of 192.168.1.20. Both have a subnet of 255.255.255.0 so they can communicate with each other.

The Pro-Watch developer will have to configure a User and a Password for Corsair to use. For this example they are “User1” and “Password1”. He will also need to determine the workstation name of the Pro-Watch server. For this example it is “DESKTOP-PWATCH”.

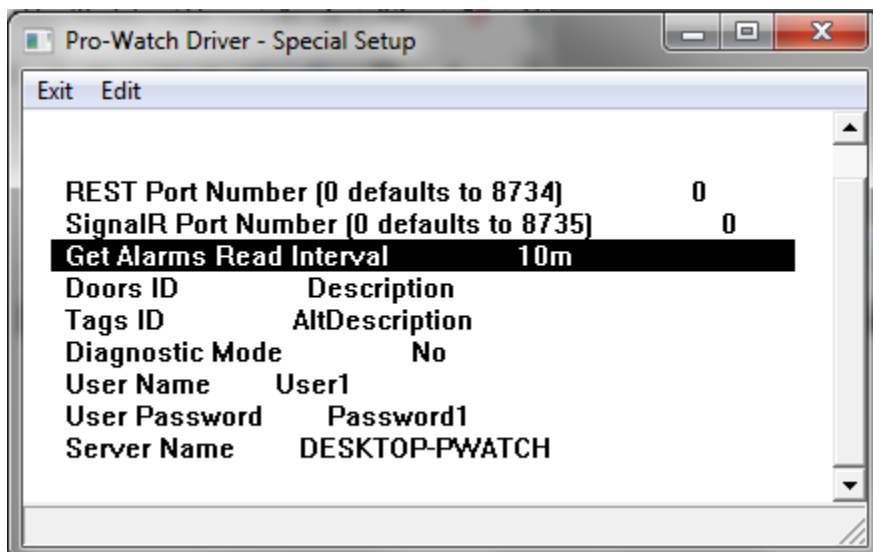
The first step for the Corsair developer is to create a driver record.



ID	Type	PLC IP
Pro-Watch Driver	ProWatch - Honeywell Pro-Watch Access Control	192.168.1.12

The driver gets a name and the Pro-Watch driver type. The IP address of the server goes into the PLC IP address field of the driver record.

Zooming on the ID field opens the special data setup for this driver.



REST Port Number (0 defaults to 8734)		0
SignalR Port Number (0 defaults to 8735)		0
Get Alarms Read Interval		10m
Doors ID	Description	
Tags ID	AltDescription	
Diagnostic Mode	No	
User Name	User1	
User Password	Password1	
Server Name	DESKTOP-PWATCH	

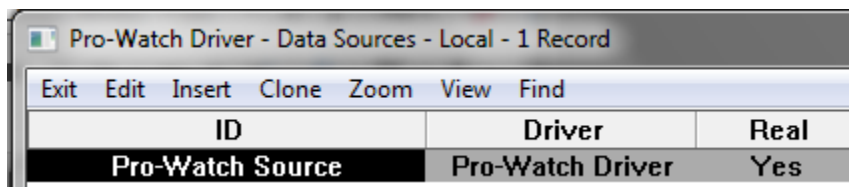
The port numbers can be left at zero if the default values are used. If the Get Alarms Read Interval is set too short Corsair can place excessive communications demand upon the Pro-Watch server. The interval should be set longer in systems with multiple Corsair computers.

Pro-Watch logical devices can be addressed in Corsair by using either their ‘Description’ or ‘AltDescription’ names. This selection can be made separately for Doors or for Tags.

The Diagnostic mode can be turned on for initial experimentation. It enables some extra capability in the driver’s Register Monitor window. Because it causes extra work for the Corsair computer it is recommended to turn it off after the system is started up, especially with a large installation.

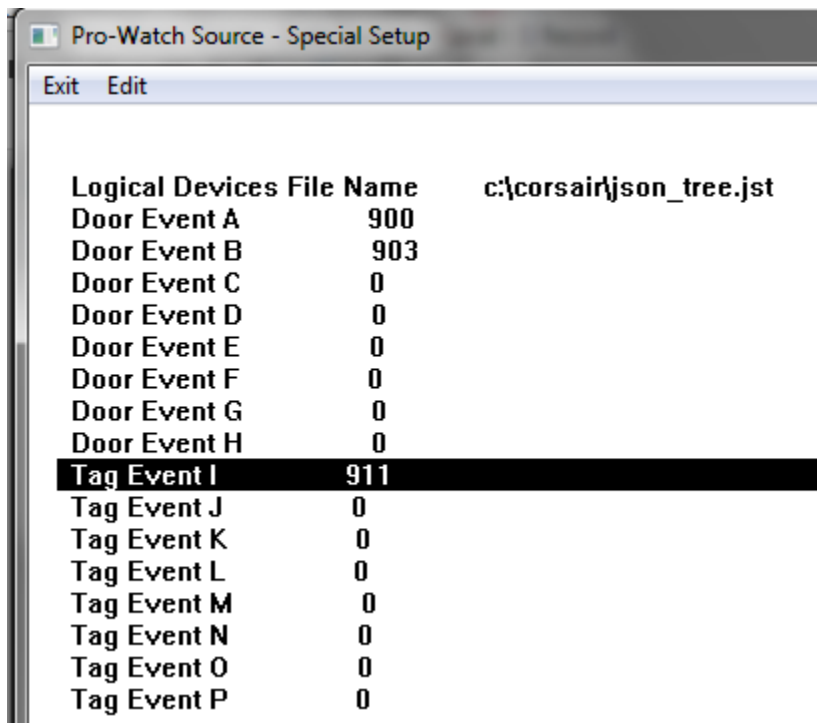
The User Name, Password, and Server Name entries must be correct to match the Pro-Watch configuration.

The next step is to close the Special Setup window, arrow to the 'Sources' field. Zoom to create a data source on the driver. Create a data source.



ID	Driver	Real
Pro-Watch Source	Pro-Watch Driver	Yes

The resulting source gets an ID name. The 'Real' field is set to 'Yes'. It does not get an IP address. Zooming on the 'ID' field opens the special data setup for the source.



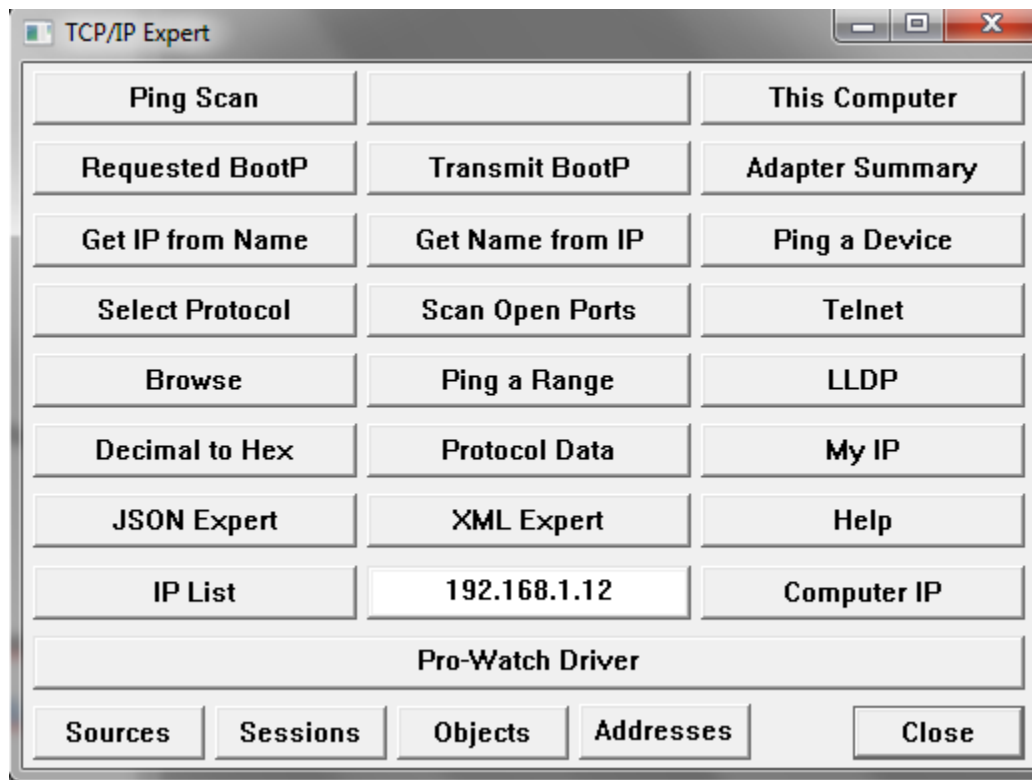
Logical Devices	File Name
	c:\corsair\json_tree.jst
Door Event A	900
Door Event B	903
Door Event C	0
Door Event D	0
Door Event E	0
Door Event F	0
Door Event G	0
Door Event H	0
Tag Event I	911
Tag Event J	0
Tag Event K	0
Tag Event L	0
Tag Event M	0
Tag Event N	0
Tag Event O	0
Tag Event P	0

The first entry is a complete file specification for the JSON tree data file that Corsair will use for the Pro-Watch Logical Device configuration information. The contents of this file will be generated later using Corsair's JSON expert. This will be done by reading the information from the Pro-Watch server through the API.

The next items are Pro-Watch Event Code numbers. The first 8 (A-H) are used to determine the secure status of doors. Corsair looks for alarm events on the logical device for the door. If there is an active alarm on any of the nonzero event codes the door is considered to be unsecure and open. In this case either a 900 or a 903 event will make the door show as unsecure.

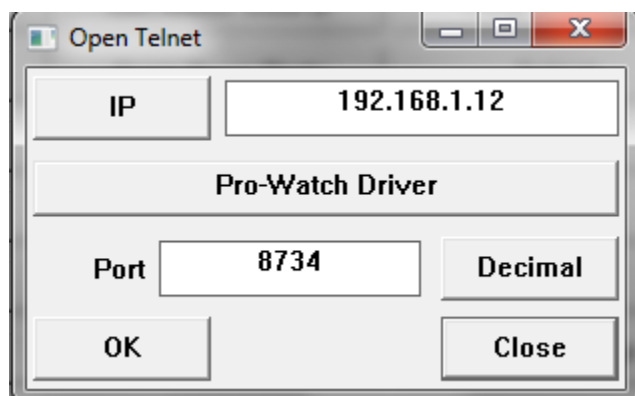
The next group of 8 (I-P) event code numbers are used to determine tag status. If any of the nonzero codes have an active alarm the tag gets a value of one. If none of them have an active alarm the tag gets a value of zero.

The Corsair developer should now save his work and proceed to the TCP Expert.

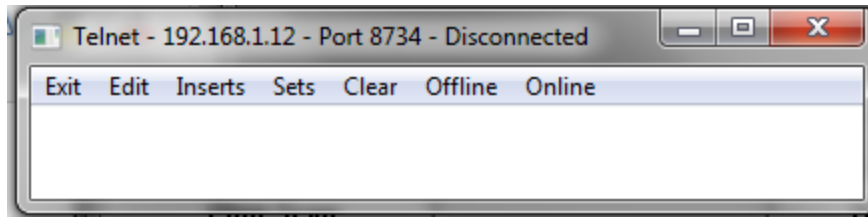


The 'IP List' button on the left can assist in entering the 192.168.1.12 address into the edit control in the center. At this point the developer can use the upper-left 'Ping Scan' button to verify that the Corsair computer can ping the Pro-Watch server. A failure to ping may indicate network equipment or Windows firewall problems.

The 'Telnet' button is now used to access Corsair's Telnet window.



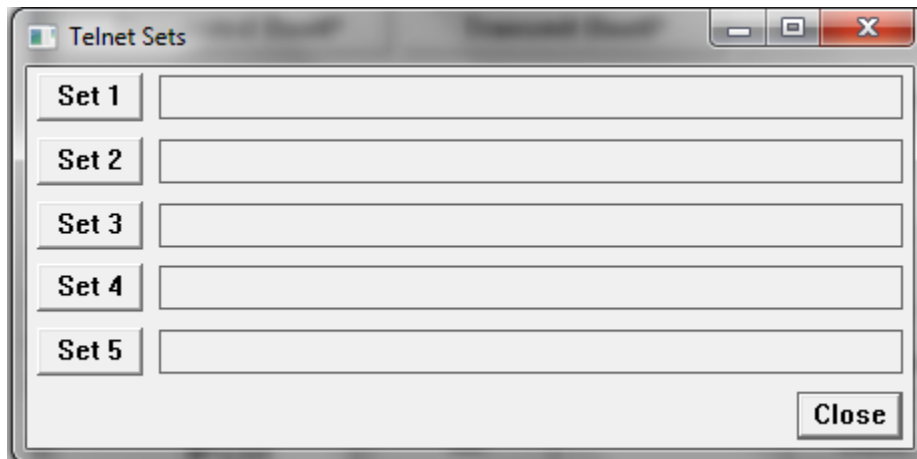
The window starts out with the default Telnet port value. The developer must change it to the 8734 that Pro-Watch uses. OK will open the Telnet window.



The 'Online' menu option should cause the window title to change to 'Connected'. 'Offline' will then return it to the disconnected state. If the ping scan was successful but Corsair fails to connect here the port numbers may not match or the Pro-Watch API is not properly started. Another possibility is firewall problems.

If the connection is successful the developer must now get the Logical Device data from the Pro-Watch server. This is done with the 'GET /logdevs' REST method. The developer could go online and type in the HTTP request manually but that would be way too complex. A better method is to construct the request using a Telnet Set. The Set can be saved in a Telnet configuration file if desired for use at another time.

The 'Edit' 'Sets' menu option opens the list of 5 available sets.



The 'Set 1' button opens the window for editing that set.

Name

Note

Verb

'Get Logical Devices' is a good name since that is what the set does. Now the components of the set need to be configured. The first button is for the Verb.

Verb ☒ GET ☐ POST ☐ PUT ☐ OPTIONS ☐ HEAD ☐ DELETE

☐ None ☐ Entry

Scheme ☒ http: ☐ None ☐ Entry

Server ☒ //IP:Port ☐ None ☐ Domain ☐ Raw

Line End ☒ HTTP/1.1 ☐ None ☐ Entry

The verb should be 'GET'. Scheme and server are set to 'None'. Now go to the first 'URL Parts' button.

☒ #1

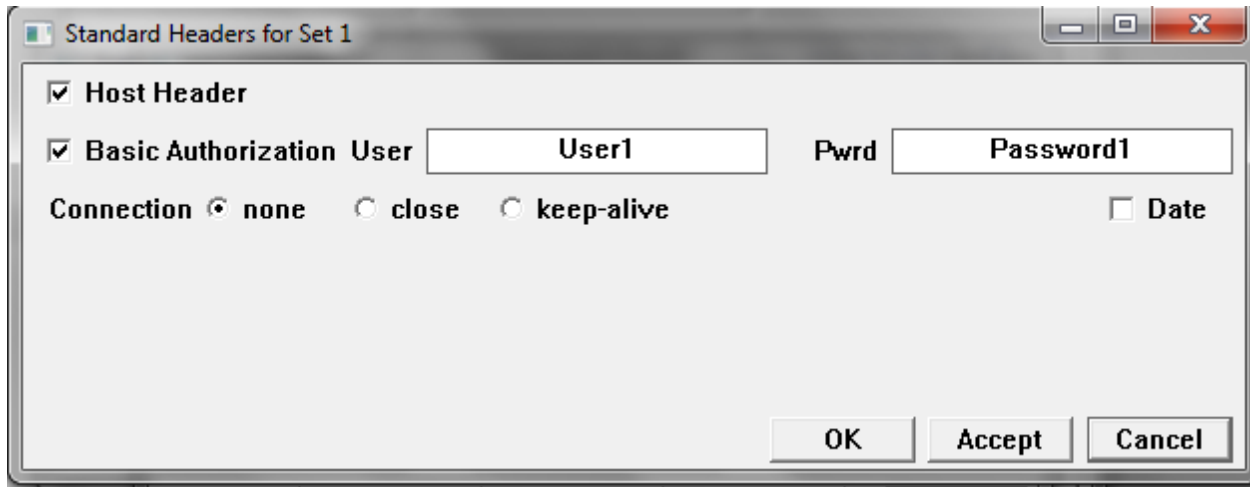
☒ #2

☐ #3

☐ #4

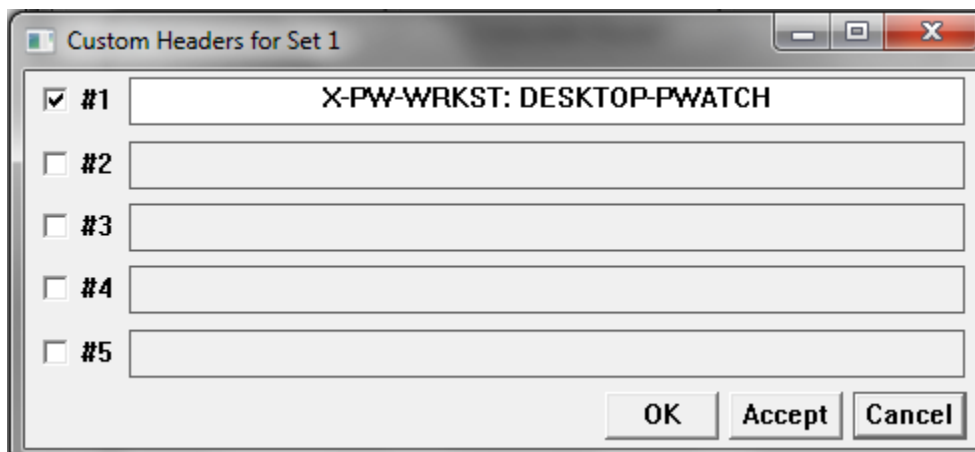
☐ #5

The two required parts are 'pwapi' and 'logdevs'. Next is the 'Standard Headers' button.



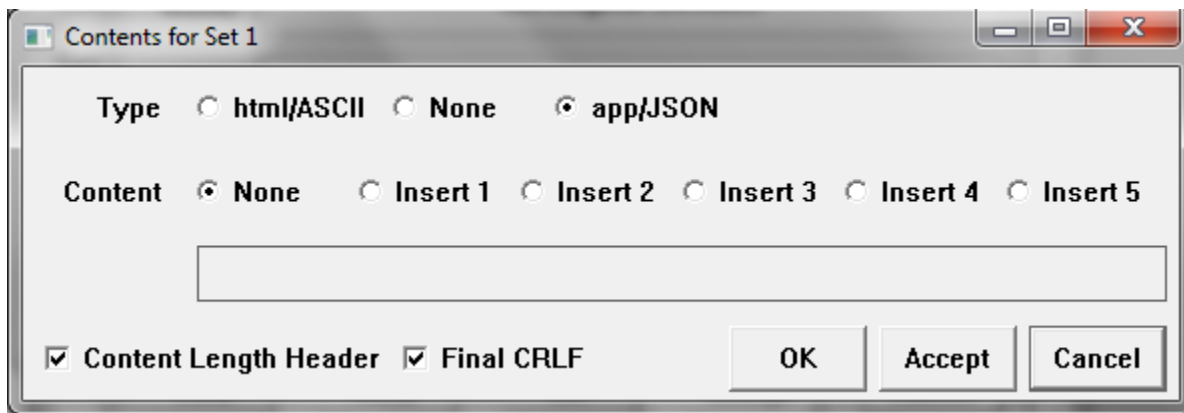
The 'Standard Headers for Set 1' dialog box is shown. It has a title bar with standard window controls. The main area contains several options: 'Host Header' is checked; 'Basic Authorization' is checked, with 'User' set to 'User1' and 'Pwrd' set to 'Password1'; 'Connection' has radio buttons for 'none' (selected), 'close', and 'keep-alive'; and a 'Date' checkbox is unchecked. At the bottom right are 'OK', 'Accept', and 'Cancel' buttons.

Host Header must be checked. The user name and password go in under Basic Authorization. Next go to the first 'Headers' button.



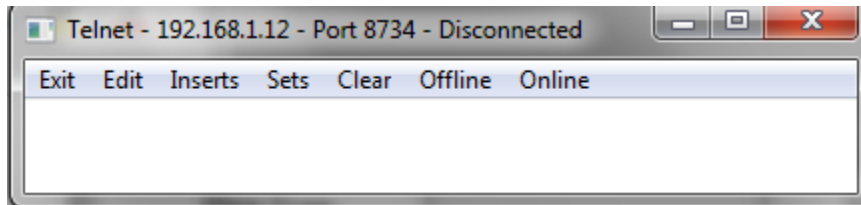
The 'Custom Headers for Set 1' dialog box is shown. It has a title bar with standard window controls. The main area contains a list of five custom headers, each with a checkbox and a text field. The first header, '#1', is checked and its text field contains 'X-PW-WRKST: DESKTOP-PWATCH'. The other headers (#2 through #5) are unchecked and have empty text fields. At the bottom right are 'OK', 'Accept', and 'Cancel' buttons.

This satisfies the Workstation Name requirement of the REST API. Now go to 'Content'.



The 'Contents for Set 1' dialog box is shown. It has a title bar with standard window controls. The main area contains two sections: 'Type' with radio buttons for 'html/ASCII', 'None', and 'app/JSON' (selected); and 'Content' with radio buttons for 'None' (selected), 'Insert 1', 'Insert 2', 'Insert 3', 'Insert 4', and 'Insert 5'. Below the 'Content' section is an empty text field. At the bottom are checkboxes for 'Content Length Header' and 'Final CRLF', both of which are checked. At the bottom right are 'OK', 'Accept', and 'Cancel' buttons.

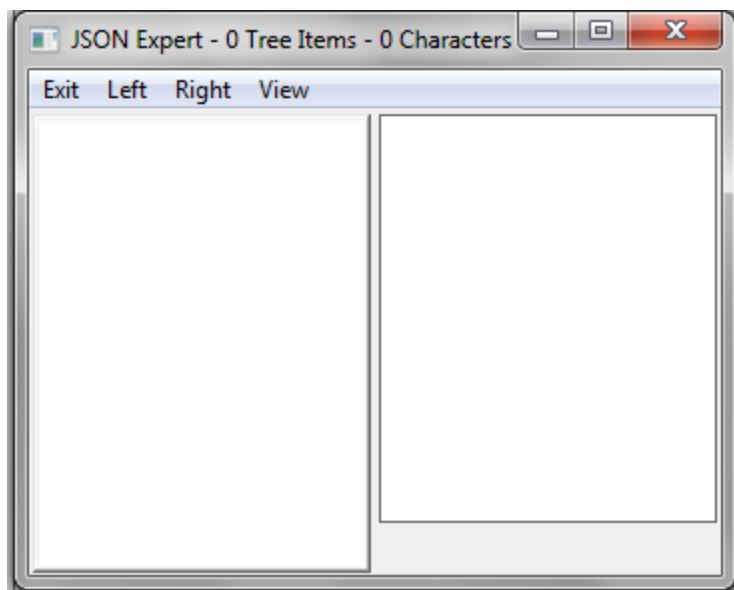
Set the type to be JSON and verify the rest. Click on OK to close this window. Close windows until you get back to the Telnet window.



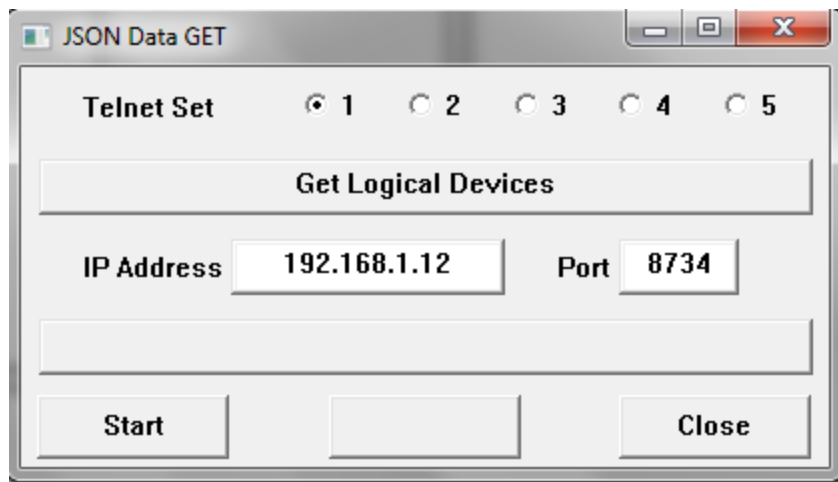
At this point it is a good idea to use the 'Edit''Save Configuration' menu option to save your telnet setup. The configuration of your Set is part of this file so you won't have to enter it again.

If you are working with a small system it may be possible to fetch the logical device data from this window. Start with a cleared window. Click on 'Online'. Then 'Sets''Trigger 1'. The content of your Set is shown in red. Pro-Watches reply is shown in back. The best reply begins with 'HTTP/1.1 200 OK'. 200 is the reply success status code. Failed replies will be short. They will have different status codes and some explanation of the problem. It may be necessary to modify the set and try again. It is recommended to go Offline, Clear the window, and then back Online for each try.

A successful reply will be quite long. Most systems with over 5 logical devices will not yield good data with this window because the reply is too long to fit in the buffer that is displayed on the screen. If you get a success reply pick the 'Edit''JSON Expert' menu option. It opens the JSON Expert window.



If the reply could fit in the available space there will be a data tree on the left side of the window. If it is blank or if it does not appear to be complete pick the 'Left''Get Data' menu option.



This JSON Data GET will load the reply from the Set directly into the JSON tree control. It can accept a very large amount of data.

Once the Logical Devices tree data appears on the left side of the JSON expert it's best to use the 'Left''Save Tree' option to save the data.

The next step is to prune unnecessary data out of the tree. Pruning will save memory and disk space but more importantly it will speed Corsair's searches of the data and make for easier monitoring in the drivers Register Monitor window.

The Logical Devices tree consists of an array of objects. Each object has several values in it. Corsair is only capable of using 3 of these values. The first is the 'LogDevID' string. This is the long difficult-to-type label that Corsair needs to interpret the messages it gets from Pro-Watch. The other fields are the 'Description' and 'AltDescription' strings. The special setup data for the driver determines which of these strings Corsair doors and tags will use. It's usually best to leave both of them in the file. The rest of the values in the objects can be pruned if desired. The pruning is accomplished with the special Delete options that are described for the JSON Expert in the Experts manual.

After the data is pruned the data should be saved in the location where the Corsair driver expects to find it. In our case it is 'c:\corsair\json_tree.jst'.

Keyscan

bbb

Doors

Corrections institutions usually have electrically operated doors. Any version of the Corsair program can be used along with a PLC to operate doors. This type of door operation is similar to what could be done

with a general-purpose interface program. Corsair offers more advanced door operations with its corrections version license option. It should be considered for any facility with more than 10 controlled doors.

Door Types

The first step for doing the PLC programming of a CorsairHMI door is to determine its type. The type determines how the F1 through F4 keys are used to control the door. It also determines the requirements for the secure indication. The secure indication is the signal from the PLC to Corsair that the door is properly closed.

The 'Monitored' type is for a door whose status is shown on the computer but it is not controlled by the PLC. Mechanically keyed doors are monitored. They have some combination of a door position and a bolt position switch to wire to a PLC input for the secure indication. The F1-F4 keys are not used for monitored doors.

The 'Half-Cycle' type door is controlled by the PLC. It is a solenoid or a motor lock without mechanical hold-back of the bolt. It uses the F1 key for an unlock button and the F2 key for a lock button. Usually F1 energizes the solenoid and F2 de-energizes it. The Half-Cycle name comes from motor locks where F1 makes the motor rotate through half its cycle and F2 finishes the rotation. The locked or unlocked status of the door must not change when the PLC power is cycled. Half-cycle locks are a frequent choice for fire egress doors. Sometimes the solenoid is energized to secure the door in these applications.

The 'Full-Cycle' type door is controlled by the PLC. They are motor or solenoid locks with mechanical hold-back. The motor runs a cycle or the solenoid energizes for a period of time. The bolt is pulled in. It stays retracted until someone pulls the door open and then shuts it. The interface has an F1-Unlock option. There is no lock option from the computer.

Full cycle locks are used in high traffic applications where the operator unlocks the door. He does not have to wait for someone to go through it before he tells the door to lock. Full cycle locks are not to be used in fire egress applications.

The 'Slider' type is used for sliding doors and gates. They are powered open or closed and capable of being stopped in mid-travel. The interface has F1-Open, F2-Close, and F3-Stop options.

PLC logic for a slider must include anti-plugging timers. Plugging a motor means taking it from open to close or close to open without allowing it to come to a stop. Plugging is damaging to electrical and mechanical components. When a motor that is opening a door stops in mid-travel the door will tend to rock open slightly and then rock back in the closed direction. If the anti-plugging timer is properly set the motor will restart in the close direction at just the right time to make for smooth motion with a minimal amount of noise and wear on the gear train. To avoid delays the programmer must take care that the anti-plug timer only is in use when it is needed. If the door is fully closed the timer is not needed to start open motion.

'Dual-Cycle' type doors are doors that can be operated in two different modes – both Full and Half cycle. The interface has F1-Unlock, F2-Lock, and F4-Hold options. F1 is used for a full-cycle operation where the door will relock by itself when closed. F4 is used for the half-cycle unlock function where the door will remain unlocked until F2 is used to lock it.

PLC operated doors are frequently interlocked in corrections applications. If two doors are in an interlock scheme only one of them can normally be opened at a time. Interlocks may need to be shut off for special situations like firefighter access.

Door Registers

PLC door data is kept in a 16-bit register. Each of the bits has a different function depending on the type of the door. Some bits act like Corsair indicators where the PLC turns them on and off and the interface displays the status. Some bits act like Corsair switches where the computer turns them on and off and the PLC uses the status. Some bits act like Corsair buttons where the computer turns them on and the PLC shuts them off. One of the most important considerations with door programming is proper behavior when the PLC power is cycled. Generally the switch type bits must be retentive. That means that whatever status they had when the power is shut off is still there when the power is turned back on. The button type bits must be shut off when the power restarts. Button actions that were performed with the PLC in Halt need to be zeroed out and ignored on the first scan. Some indicators need to be retentive and some are not.

A programmer using a PLC with Modbus style addresses may place the first door at bit 00001 through bit 00016. The second door may take bit 00017 through bit 00032. Another possibility is for the first door to use Holding Register 40001. The bits would be 40001/0 to 40001/15. The second door could go into Holding Register 40002. Input Registers like 30001 cannot be used for doors since the computer cannot write data to them. Inputs bits like 10001 cannot be used for the same reason.

The Modbus protocol does not specify the power cycling retention rules for different data types. It is up to the PLC manufacturer so their documentation must be consulted for retention rules. Frequently the door status data is held in retentive memory. The programmer defines a single 'Power-Up' boolean variable. It comes up False (0) on the first scan when the PLC starts. 30 seconds later it is set to True (1). Door operations are not allowed until Power-Up is true. This allows time for remote I/O and peer-to-peer communications to get started first. This use of the Power-Up signal can help guarantee that the PLC follows the proper memory retention rules.

Another concern for the PLC programmer is the method that CorsairHMI must use to change the bit data of the type that he has chosen for the door register. Data in Modbus 0#### coils can be written on an individual coil basis. Corsair can turn a button on without changing any of the other bits in the register. 4#### holding registers are a different issue. Some PLC's support a bitmasked write command where Corsair sends a pattern of 1s and 0s that the PLC uses to only change the desired bit of the register. If a PLC does not support this command Corsair must use a Read-Modify-Write (RMW) operation. It reads the holding register and then writes that value back to the PLC with the desired bit changes. Problems can occur if the PLC logic changes bits in the short time between when Corsair reads

the value and then writes it back. Any PLC logic, including the samples given here, must be evaluated by the PLC programmer to see that it will always work correctly with a RMW operation. Bitmasked writes should be used whenever possible to avoid issues and speed up door operations.

This document refers to the bits of the door status register as bit 0 through bit 15. Bit 0 is the least significant bit and bit 15 is the most significant. If a door is located at Modbus coil addresses 00017 through 00032 coil 00017 is bit 0 and coil 00032 is bit 15. A door located at Modbus Holding Register address 40010 has 40010/0 for bit 0 and 40010/15 for bit 15.

Each of the 16 different bits of a door register has a name and function that depends on the door type. There are some general rules that can be considered before looking at each of the 5 door types. In cases where a bit is supposed to always be on or off the register bits are not to be trusted as they can be changed from external data communications. PLC logic should be written to force these bits to always be the correct value.

Bit 0 is always used as the 'Secure' bit. It is an indicator that is turned on and off by the PLC. It should be on only if all conditions securing the door are true. This bit is what Corsair uses to show the status of the door on the screen. The programming for the Secure bit varies with the type of the door. It is usually at the end of the logic. The Monitored door is the only type where the Secure bit is only dependent on the Door Position Switch indication. For all the PLC-controlled door types the door is secure if the switch is correct and the PLC is not trying to open the door. This will help to prevent false secure indications when switches are defective.

Bit 1 is always used as the 'DPSI' 'Door Position Switch Indication' bit. It acts as an indicator. Most door hardware provides a signal to a PLC input that shows the door is secure. This signal is usually from more than one switch wired in series. There may be both a door position and a bolt position switch. The DPSI indicator is controlled from one or more DPS PLC inputs in a series (AND) configuration. This is usually at the beginning of the logic. The DPSI bit is used as a part of the logic for the status of the Secure bit.

Bit 2 is sometimes used as the 'P_Cust' 'Protective Custody' bit. It acts like a switch. The computer turns it on (1) for the door to have protective custody status. The operator will then have to answer a special prompt to open the door. If the computer turns the bit off (0) the door can be operated normally. The status of this bit must be retained through a power failure. Some projects may require a high-security door to always have protective custody. Other doors should never have that status. In these cases PLC logic should explicitly force the bit on or off.

Bit 3 is sometimes used as the 'Access' 'Inmate Access' switch. If it is on the inmate can open his cell door through some means. If it is off the inmate cannot open the door. It must be retentive.

Bit 4 is sometimes used as the 'CTRL' 'Interface Control' switch. The use of this switch is entirely up to the PLC programmer. The Corsair program can turn it on and off but it makes no assumptions about what the switch is used for. Cell lighting is a possibility. It may or may not retain at the discretion of the PLC programmer.

Bit 5 is sometimes used as the 'ILockO' 'Interlock On/Off' switch. If the computer turns it on (1) the door's interlocks are active and it cannot be opened until other doors are secure. If the computer turns it off (0) the interlock has been overridden and the door can be opened. It must be retentive. Some high-security applications may require this bit to be forced on in PLC logic so the interlock cannot be shut off. Firefighter access must be considered in making this decision.

Bit 6 is sometimes used as the 'ILockP' 'Interlock Permissive' indication. The PLC turns this bit on if all other doors in the interlock scheme are secure so the door may be opened. If the ILockO bit is off the ILockP bit is turned on. If a door is not interlocked ILockP should be turned on all the time.

Assume that doors A, B, C, and D are all interlocked so only one can be opened at a time. Ladder logic for the ILockP on door A would look as follows:

```
Secure_B      Secure_C      Secure_D      ILockP_A
---| |-----| |-----| |-----+------( )-----
ILockO_A                      |
---|/|-----+-----
```

If door E is not interlocked the ILockP bit should be forced on.

```
ILockP_E
------( )-----
```

Bits 8, 9, 10, and 11 are used for the F1, F2, F3, and F4 keys. The purpose varies with door type but in each case they act like button tags. Corsair turns the bit on when the operator hits the key and the PLC turns it off after performing the desired action.

Any of the 4 button bits that are used for a lock need to perform their action if and only if the Power_Up signal is true. After examining the bits the PLC should always shut them off.

```
Button
------( U )-----
```

Keys hit before Power_Up will be ignored. This effectively makes the buttons nonretentive.

Bits 12 and 13 are reserved for future use by the Corsair program. They should not be used for any purpose.

There is a Door data maintenance popup available from an operator screen. Hook the cursor to a door placement and press the 'M' maintenance key.

Half-Cycle Cell Door				
Door PLC				
Half-Cycle				
<input type="checkbox"/>	Secure	000001	<input type="checkbox"/> DPSI	000002
<input type="checkbox"/>	P_Cust	000003	<input type="checkbox"/> Access	000004
<input type="checkbox"/>	CTRL	000005	<input type="checkbox"/> ILock0	000006
<input checked="" type="checkbox"/>	ILockP	000007	<input type="checkbox"/> Unused1	000008
<input type="checkbox"/>	Unlk_B	000009	<input type="checkbox"/> Lock_B	000010
<input type="checkbox"/>	Button3	000011	<input type="checkbox"/> Button4	000012
<input type="checkbox"/>	Future1	000013	<input type="checkbox"/> Future2	000014
<input type="checkbox"/>	Hold_Open	000015	<input type="checkbox"/> Unused2	000016

F1 F2 F3 F4 Close

The maintenance window shows the name of the door, what PLC it is on, its type, the address of each door register bit and its current on-off status.

The following sections give specific information as to the bit assignments for each of the door types. Some suggestions for PLC logic are included. In every case the PLC programmer must assume final responsibility for what the PLC does. Lockdown and Fire Egress must be taken into account.

Monitored Doors

Bit 0 – Secure – Indicator

Bit 1 – DPSI – Door Position Switch Indication - Indicator

Bit 2 – P_Cust – Protective Custody – Switch Do not use on a monitored door.

Bit 3 – Access – Switch

Bit 4 – CTRL – Interface Control – Switch

Bit 5 – Unused1 – Switch Use for any purpose

Bit 6 – Unused2 – Available for any use

Bit 7 – Unused3 – Available for any use

Bit 8 – Button1 – F1 Key – Button Do not use

Bit 9 – Button2 – F2 Key – Button Do not use

Bit 10 – Button3 – F3 Key – Button Do not use

Bit 11– Button4 – F4 Key – Button Do not use

Bit 12 - Future1 - Do not use Reserve for future versions of Corsair

Bit 13 - Future2 - Do not use Reserve for future versions of Corsair

Bit 14 – Unused4 – Available for any use

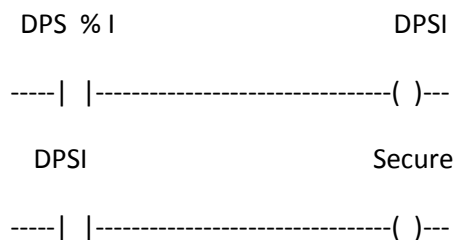
Bit 15 – Unused5 – Available for any use

Pseudo Code:

DPSI = DPS; (* Copy DPS input into the DPSI Indicator *)

SECURE = DPSI; (* Copy the DPSI Indicator into the Secure bit *)

Ladder:



The door is secure when the switch is on.

Half-Cycle Doors

Bit 0 – Secure – Indicator

Bit 1 – DPSI – Door Position Switch Indication - Indicator

Bit 2 – P_Cust – Protective Custody – Switch

Bit 3 – Access – Switch

Bit 4 – CTRL – Interface Control – Switch

Bit 5 – ILockO –Interlock On/Off - Switch

Bit 6 – ILockP –Interlock Permissive – Indicator

Bit 7 – Unused1 – Available for any use

Bit 8 – Unlk_B – Unlock F1 Key – Button

Bit 9 – Lock_B - Lock F2 Key – Button

Bit 10 – Button3 – F3 Key – Button Do not use

Bit 11– Button4 – F4 Key – Button Do not use

Bit 12 - Future1 - Do not use Reserve for future versions of Corsair

Bit 13 - Future2 - Do not use Reserve for future versions of Corsair

Bit 14 – Hold_Open – Indication

Bit 15 – Unused2 – Available for any use

The Hold-Open bit should be retentive, especially for a fire egress door.

Pseudo Code:

DPSI = DPS; (* Copy DPS input into the DPSI Indicator *)

Calculate ILockP

If Unlk_B AND ILockP and Power_Up – Turn on Hold_Open

IF Lock_B AND Power_Up – Turn off Hold_Open

Reset Unlk_B to 0

Reset Lock_B to 0

Copy Hold_Open to External Output %Q

SECURE = DPSI AND (NOT Hold_Open;

Ladder

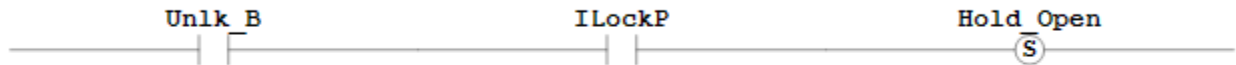
Copy External Input into DPSI - On if Secure



This door is not interlocked



Unlock if the Interlock is OK (Retentive)



Lock from the F2 Lock Button



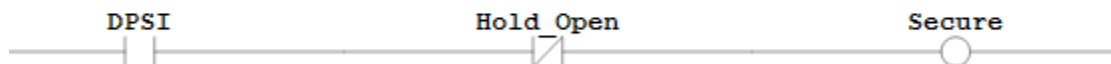
Buttons are shut off by the plc and are not retained on CPU start



Copy Hold_Open to External Output



The door is secure when the switch is on and the output is off



Full-Cycle Doors

Bit 0 – Secure – Indicator

Bit 1 – DPSI – Door Position Switch Indication - Indicator

Bit 2 – P_Cust – Protective Custody – Switch

Bit 3 – Access – Switch

Bit 4 – CTRL – Interface Control – Switch

Bit 5 – ILockO –Interlock On/Off - Switch

Bit 6 – ILockP –Interlock Permissive – Indicator

Bit 7 – Unused1 – Available for any use

Bit 8 – Unlk_B – Unlock F1 Key – Button

Bit 9 – Button2 - F2 Key – Button Do not use

Bit 10 – Button3 – F3 Key – Button Do not use

Bit 11– Button4 – F4 Key – Button Do not use

Bit 12 - Future1 - Do not use Reserve for future versions of Corsair

Bit 13 - Future2 - Do not use Reserve for future versions of Corsair

Bit 14 - Cycle – Indicator

Bit 15 – Unused2 – Available for any use

With a full-cycle lock the output is held on for a fixed period of time. Full cycles may be solenoids or motors. If it is a solenoid the timing must be long enough to make sure the bolt is pulled solidly into the mechanical holdback. Motor Full Cycles sometimes use a microswitch on a cam to keep the motor running through a complete cycle. The minimum and maximum times for the output need to be determined. To get the minimum time reduce the timing until the lock occasionally does not do a complete cycle since the motor doesn't have enough movement to get to the latching microswitch. To get the maximum time increase the timing until the motor occasionally does two complete cycles instead of one. The correct time to use for the setpoint is halfway between this minimum and maximum.

Pseudo Code:

DPSI = DPS; (* Copy DPS input into the DPSI Indicator *)

Calculate ILockP

If Unlk_B AND ILockP AND Power_Up – Turn on Cycle

If Time - Turn off Cycle

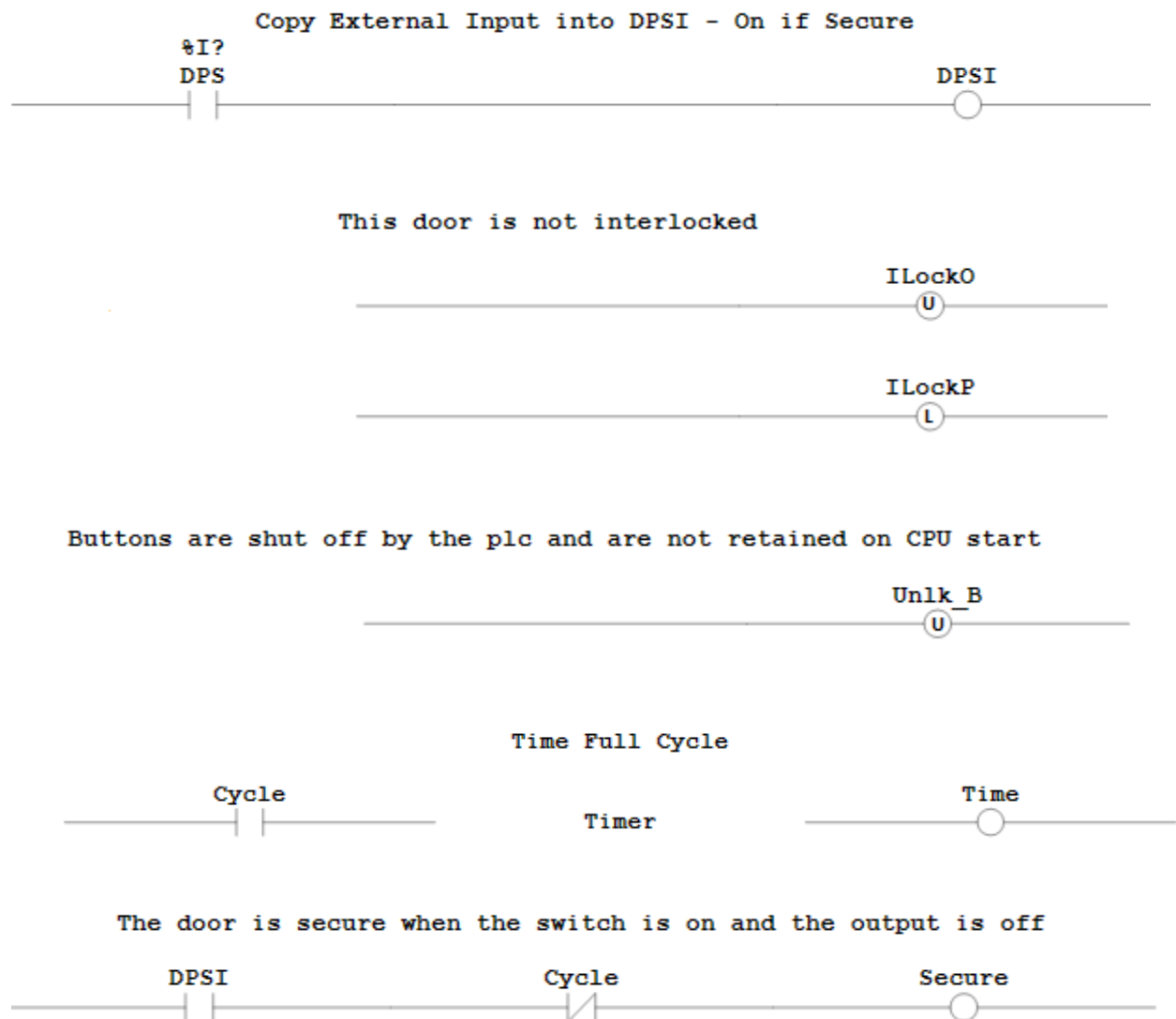
On-Delay Timer – Input is Cycle, Output is Time

Reset Unlk_B to 0

Copy Cycle to External Output %Q

SECURE = DPSI AND (NOT Cycle);

Ladder: (Not Complete)



Sliders

Bit 0 – Secure – Indicator

Bit 1 – DPSI – Door Position Switch Indication - Indicator

Bit 2 – P_Cust – Protective Custody – Switch

Bit 3 – Access – Switch

Bit 4 – CTRL – Interface Control – Switch

Bit 5 – ILockO –Interlock On/Off - Switch

Bit 6 – ILockP –Interlock Permissive – Indicator

Bit 7 – Open_Limit – Optional Door Fully Open Indication

Bit 8 – Open_B – Open F1 Key – Button

Bit 9 – Close_B - Close F2 Key – Button

Bit 10 – Stop_B – Stop F3 Key – Button

Bit 11– Button4 – F4 Key – Button Do not use

Bit 12 - Future1 - Do not use Reserve for future versions of Corsair

Bit 13 - Future2 - Do not use Reserve for future versions of Corsair

Bit 14 – Opening – Indication

Bit 15 – Closing – Indication

Both the Opening and Closing indications need to go to zero when Power_Up is not on.

Many times sliders and gates do not have full-open switches wired to the PLC. The Open output is energized for a fixed time period in these cases. The Close output is shut off when the closed limit switch is reached. It needs to also be timed off in case the limit switch is defective. Outputs without timeouts may be dangerous for maintenance people working on the door as they could cause unexpected door operation. Timeouts should be set for the normal full travel time plus a few seconds. Pneumatic doors may have more variations in travel times than motorized doors so they should get longer timeouts.

Pseudo Code:

DPSI = DPS; (* Copy DPS input into the DPSI Indicator *)

SECURE = DPSI; (* Copy the DPSI Indicator into the Secure bit *)

Ladder: (Not complete)

Copy External Input into DPSI - On if Secure



Optional - Copy External Input into Open_Limit - On if Open



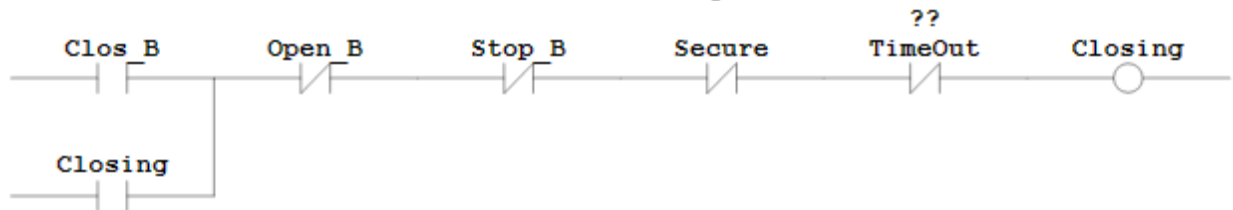
This door is not interlocked

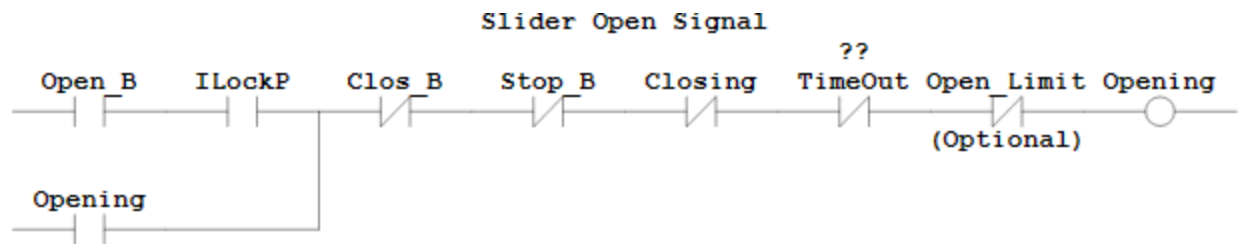


Time travel in either direction

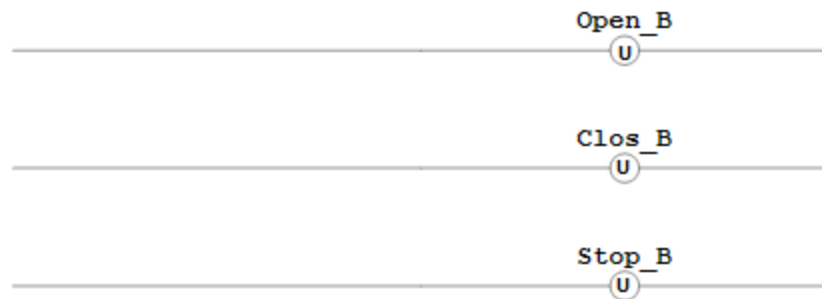


Slider Close Signal

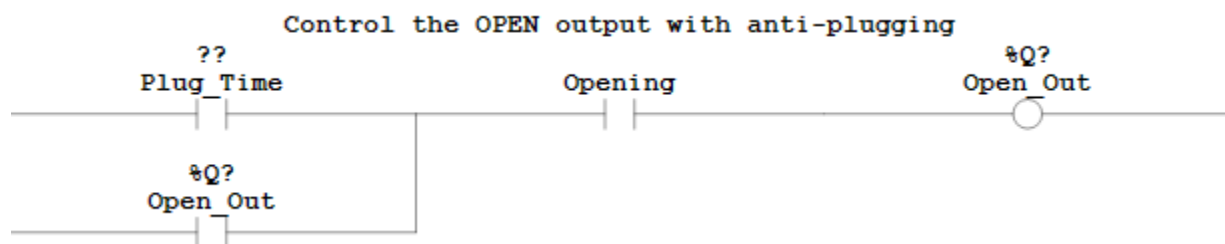
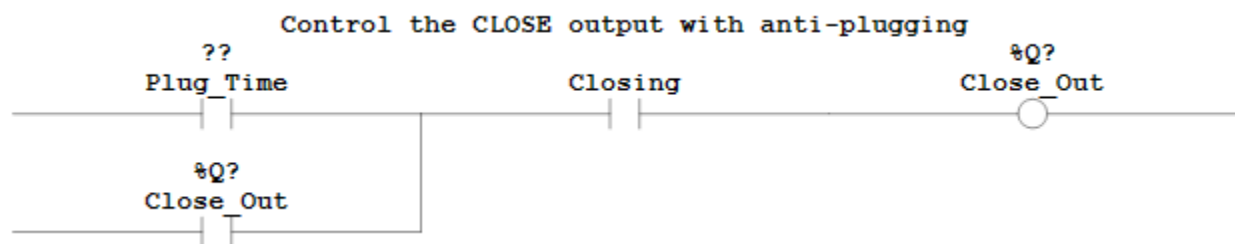
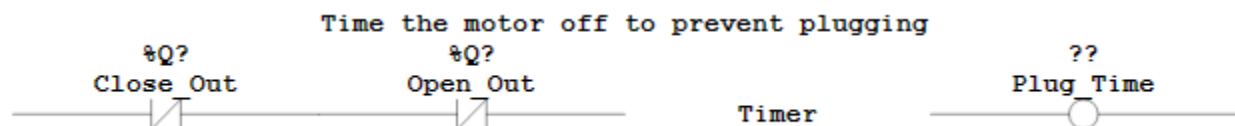




Buttons are shut off by the plc and are not retained on CPU start



DPS % I



The door is secure when the switch is on and it is not opening



Dual-Cycle Doors

Bit 0 – Secure – Indicator

Bit 1 – DPSI – Door Position Switch Indication - Indicator

Bit 2 – P_Cust – Protective Custody – Switch

Bit 3 – Access – Switch

Bit 4 – CTRL – Interface Control – Switch

Bit 5 – ILockO –Interlock On/Off - Switch

Bit 6 – ILockP –Interlock Permissive – Indicator

Bit 7 – Cycle – Full Cycle Indication

Bit 8 – Unlk_B – Unlock F1 Key – Button

Bit 9 – Lock_B - Lock F2 Key – Button

Bit 10 – Button3 – F3 Key – Button Do not use

Bit 11– Hold_B – Hold F4 Key – Button

Bit 12 - Future1 - Do not use Reserve for future versions of Corsair

Bit 13 - Future2 - Do not use Reserve for future versions of Corsair

Bit 14 – Hold_Open – Indication

Bit 15 – Unused1 – Available for any use

The timing guidelines for a dual-cycle door are the same as for a full-cycle.

Pseudo Code:

DPSI = DPS; (* Copy DPS input into the DPSI Indicator *)

Calculate ILockP

If Unlk_B AND ILockP AND Power_Up – Turn on Cycle

If Time – Turn off Cycle

On-Delay Timer – Input is Cycle, Output is Time

If Hold_B AND ILockP AND Power_Up – Turn on Hold_Open

If Lock_B – Turn off Hold_Open

Reset Unlk_B to 0

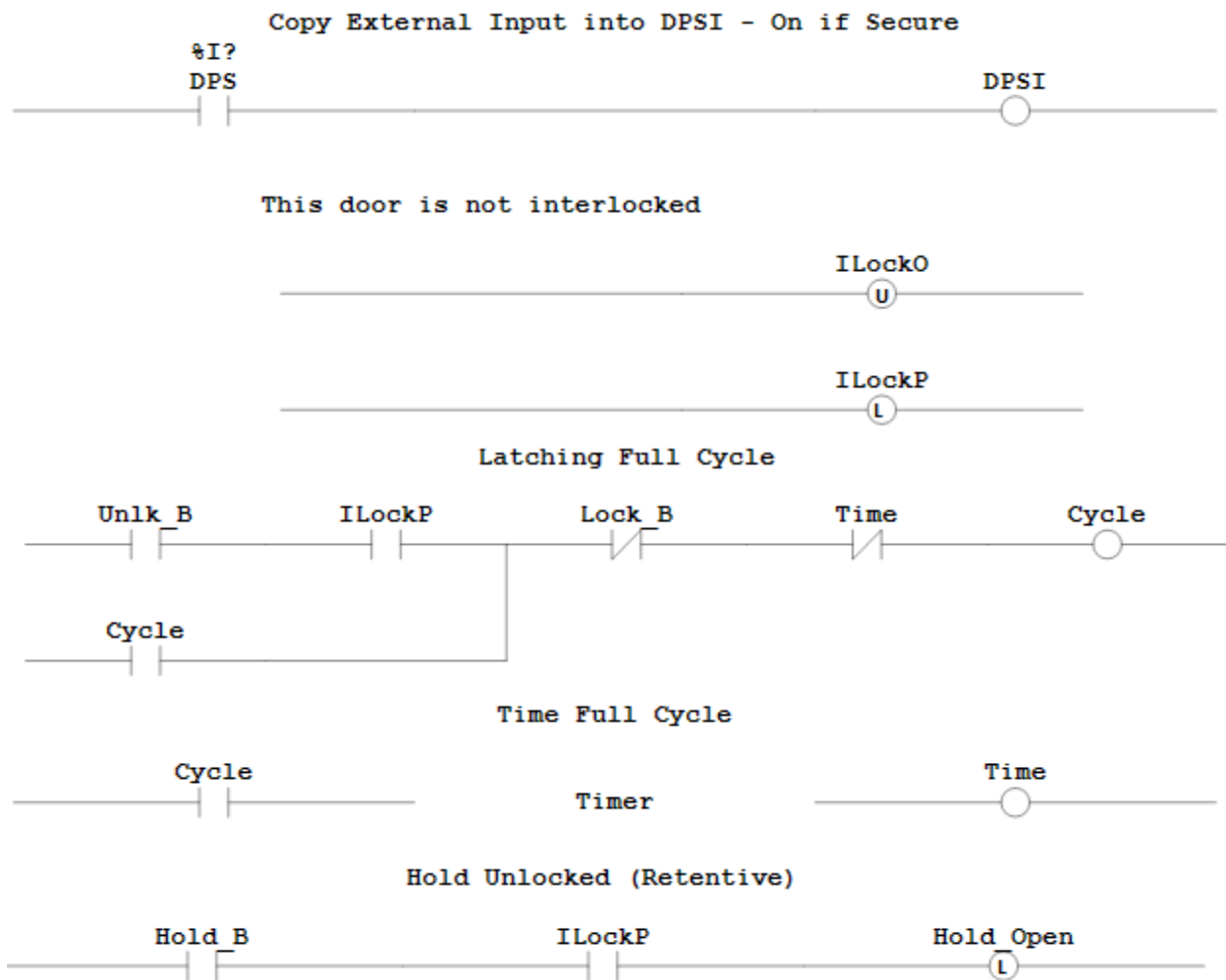
Reset Lock_B to 0

Reset Hold_B to 0

External Output %Q = Cycle OR Hold_Open

SECURE = DPSI AND (NOT Cycle) AND (NOT Hold_Open);

Ladder: (Not complete)



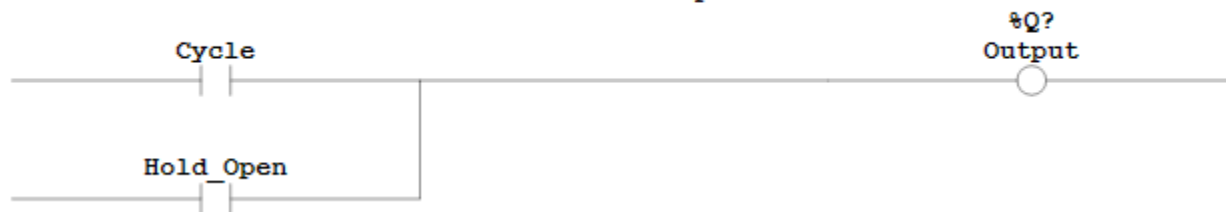
Lock - Release Hold



Buttons are shut off by the plc and are not retained on CPU start



External Output



The door is secure when the switch is on and the output is off



Group Operation

This document does not specify how doors are operated in groups as there are many variations in the required sequence of operation. Some systems will require doors with Protective Custody status to not open with group opens. They should only group open if the P_Cust bit is off.

Time delay staggering of openings on grouped doors is a frequent requirement to avoid electrical or pneumatic inrush problems when many locks are operated. A large solenoid lock can draw inrush current for a few seconds. Opening a number of doors at a time could overload power supplies, blow fuses, or drop the pressure on an air compressor. The PLC programmer should utilize timing to minimize the possibility of this happening. This is especially important for group openings of doors in a Fire Egress situation.