Honeywell Panel Bus Driver for Tridium Niagara

User Guide

Contents

1 Introduction

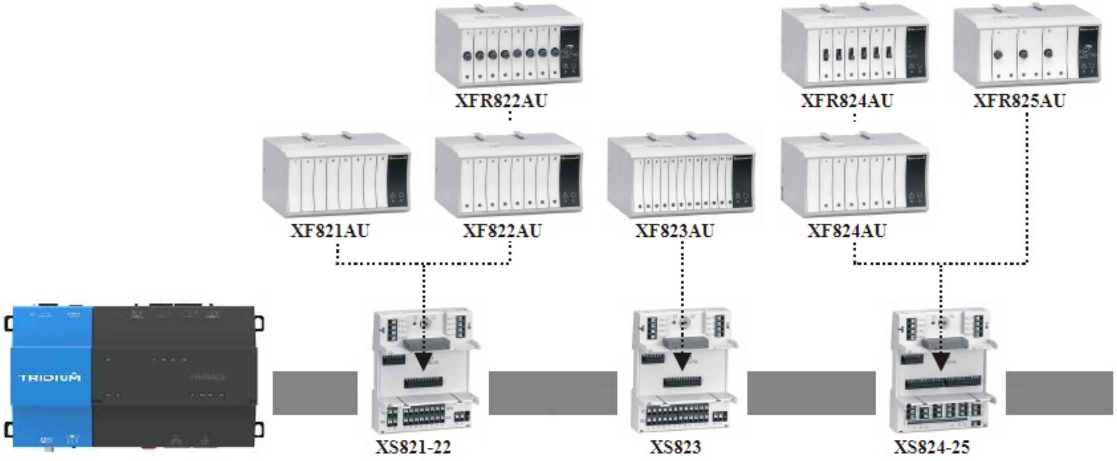
Honeywell Panel Bus is a protocol for IO modules for Honeywell controllers, such as Excel 800, Excel Web II, ComfortPoint Open Plant, LION, EAGLE and EAGLEHAWK [1]. The family of these IO modules includes multiple pluggable modules with just one type of IO (analog input, binary input, analog output, relay output and floating / three-position output) or mixture of IO.

Model	AI	BI	AO	RO	FO	Override
XF821A / CLIOP821A	8					
XF822A / CLIOP822A			8			
XFR822A / CLIOPR822A			8			X
XF823A / CLIOP823A		12				
XF824A / CLIOP824A				6		
XF824A / CLIOPR824A				6		X
XFR825A / CLIOPR825A					3	X
XF830A / CLIOP830A	8	12	8	6		
CPO-IO830 / CLIOP831A	8	12	8	6		

Panel Bus modules communicate over RS-485 interface using 115.2 kbps baud rate. They are hot-pluggable – no need to disconnect power to swap them. All addressing is done using one HEX switch. Output modules are available with manual override switches / potentiometers for easy maintenance. Panel Bus modules allow to cover all kinds of applications, which require from tens to hundreds IO points.

For a full description of Panel Bus refer to Panel Bus I/O Modules.

Tridium Niagara driver allows to retrofit Honeywell panels by replacing main controller with JACE or any other Niagara-compatible device (Cylon Integra, Johnson Controls FX, Distech Controls EC-BOS, GC5 iSMA-B-MAC36NL etc.) and keeping all IO modules. That greatly reduces the amount of work and price of the new controller IO. Panel Bus IO points will become available for control by Niagara logic on a standard wiresheet. Each IO point can also be configured, e.g. set thermistor types for inputs or fail-safe value for outputs.



[1] All trademarks or registered trademarks are property of their respective owners

2 Requirements

- Niagara-powered device such as Jace 8000 or any OEM version
- Jace RS-485 port or serial-to-IP device with Baudrate Virtual Port driver
- Panel Bus driver module and license

3 Quick Start

1. Import panelBus.jar (AX) or panelBus-rt.jar (N4) module with all dependencies into both Jace and WorkPlace, restart *both*.
2. Add **Panel Bus Network** to the station.
3. Enter license code in **License** property under **Panel Bus Network**.
4. Set COM port under **Serial Config / Port Name**. Make sure port settings are correct (Panel Bus default settings are 115200/8/1/None).
5. **Discover** devices – you should be able to see IO modules connected to the COM port.
6. Add discovered devices to the station.
7. Open device **Points** extension and **Discover** points.
8. Add points to the station. Points shall show their values and output signals could be changed.

4 Panel Bus Network

Panel Bus Network contains many standard Niagara properties, as well as few driver-specific ones:

- **License** – the code which allows the driver to run on your host
- **Tuning Policies / Default Policy / Read IO Fault Status** – whether point status shall be read from IO module and applied to Niagara point. By default it is set to false.
- **Tuning Policies / Default Policy / Always Subscribed** – whether the point shall always be subscribed: periodically read from IO module.

Normally only **License** and **Serial Config / Port Name** shall be changed. It is possible to have multiple **Panel Bus Network** in Niagara station, each with its own COM port. Points in all networks are counted for licensing purposes.

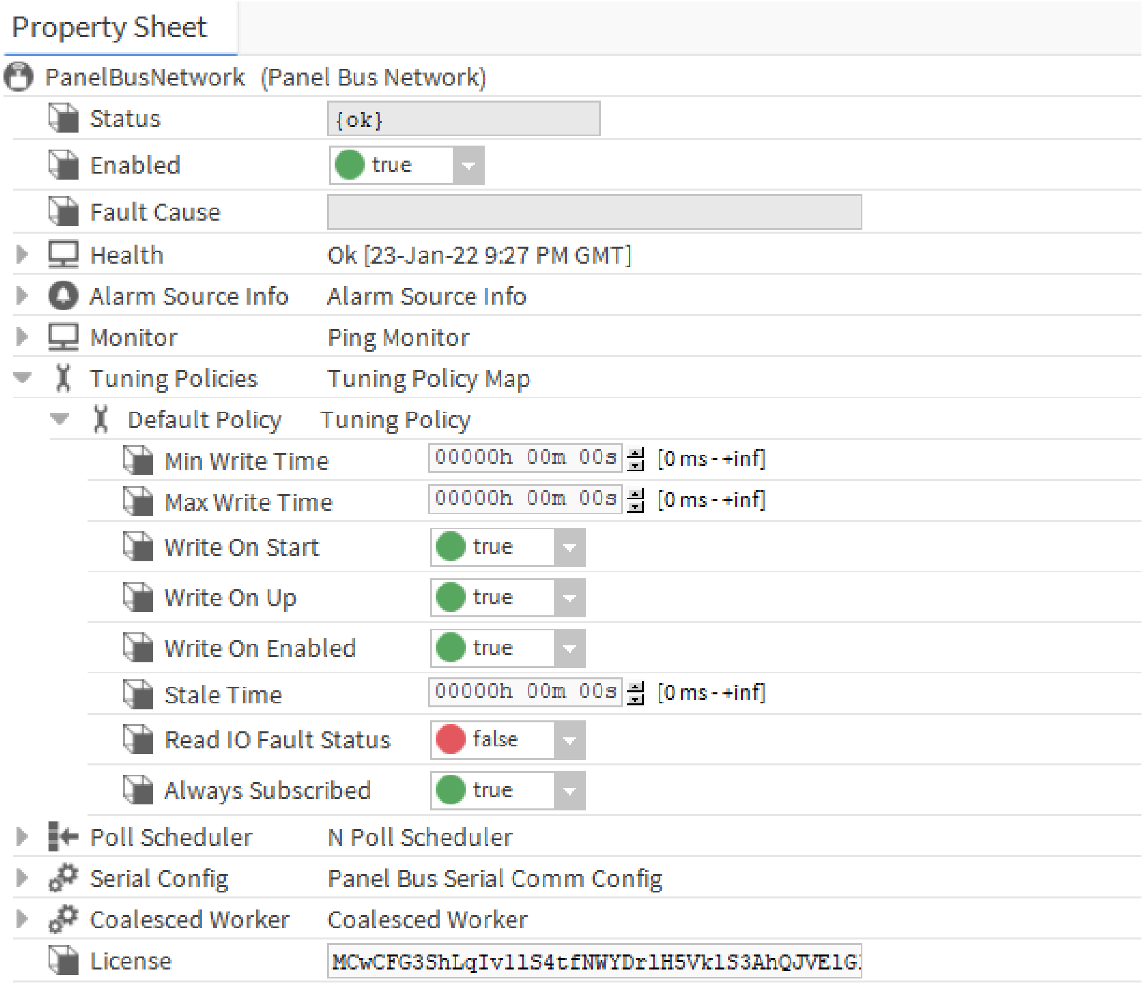
5 Panel Bus Device

Panel Bus IO modules are addressed using *S2 HEX switch* on them, so each module has an address expressed as one hexadecimal digit for 0 to F. Each type of IO modules (AI, BI, AO, BO, mixed) is addressed separately. For example it is possible to have one CLIOP821A module with address 0 and one CLIOP822A with address 0 on the same line.

Panel Bus Device contains the following driver-specific properties:

- **Model** – IO device model
- **Address** – Hexadecimal number set by S2 switch on IO module
- **Serial** – Device serial number, read only
- **Firmware** – Device firmware version, read only
- **Poll Frequency** – Frequency of point polling, either *Fast*, *Normal* or *Slow*. Timings are specified in **PanelBusNetwork / Poll Scheduler**.

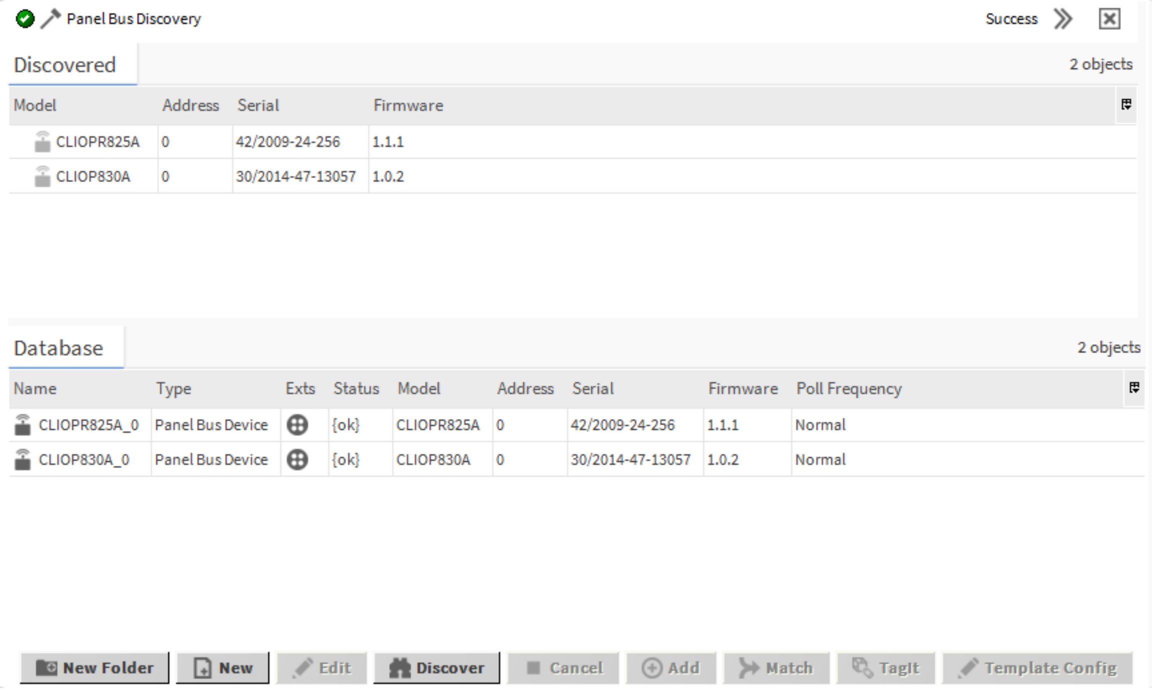
Devices can be discovered automatically or added manually. In case of automatic discovery no properties shall be changed.



PanelBusNetwork (Panel Bus Network)

Status	{ok}
Enabled	● true
Fault Cause	
Health	Ok [23-Jan-22 9:27 PM GMT]
Alarm Source Info	Alarm Source Info
Monitor	Ping Monitor
Tuning Policies	Tuning Policy Map
Default Policy	Tuning Policy
Min Write Time	00000h 00m 00s [0 ms - +inf]
Max Write Time	00000h 00m 00s [0 ms - +inf]
Write On Start	● true
Write On Up	● true
Write On Enabled	● true
Stale Time	00000h 00m 00s [0 ms - +inf]
Read IO Fault Status	● false
Always Subscribed	● true
Poll Scheduler	N Poll Scheduler
Serial Config	Panel Bus Serial Comm Config
Coalesced Worker	Coalesced Worker
License	MCwCFG3ShLqIvllS4tfNWYDrlH5VklS3AhQJVElG

Figure 1: Panel Bus network properties



Discovered 2 objects

Model	Address	Serial	Firmware
CLIOPR825A	0	42/2009-24-256	1.1.1
CLIOP830A	0	30/2014-47-13057	1.0.2

Database 2 objects

Name	Type	Exts	Status	Model	Address	Serial	Firmware	Poll Frequency
CLIOPR825A_0	Panel Bus Device	⊕	{ok}	CLIOPR825A	0	42/2009-24-256	1.1.1	Normal
CLIOP830A_0	Panel Bus Device	⊕	{ok}	CLIOP830A	0	30/2014-47-13057	1.0.2	Normal

Figure 2: Panel Bus devices

6 Panel Bus Points

Panel Bus points are discovered automatically in **Panel Bus Point Manager**. Each point has the following properties:

- **Type** – Niagara point type, Boolean or Numeric Point for inputs and Boolean or Numeric Writable for outputs
- **Facets** – Point units
- **Point Type** – Panel Bus IO type
- **Address** – IO number
- **IO Type** – Panel Bus IO configuration.

Point Type and **Address** are automatically assigned when points are added to the station from the discovery pane. **Type** is also automatically selected based on **Point Type**, but it could be changed if needed.

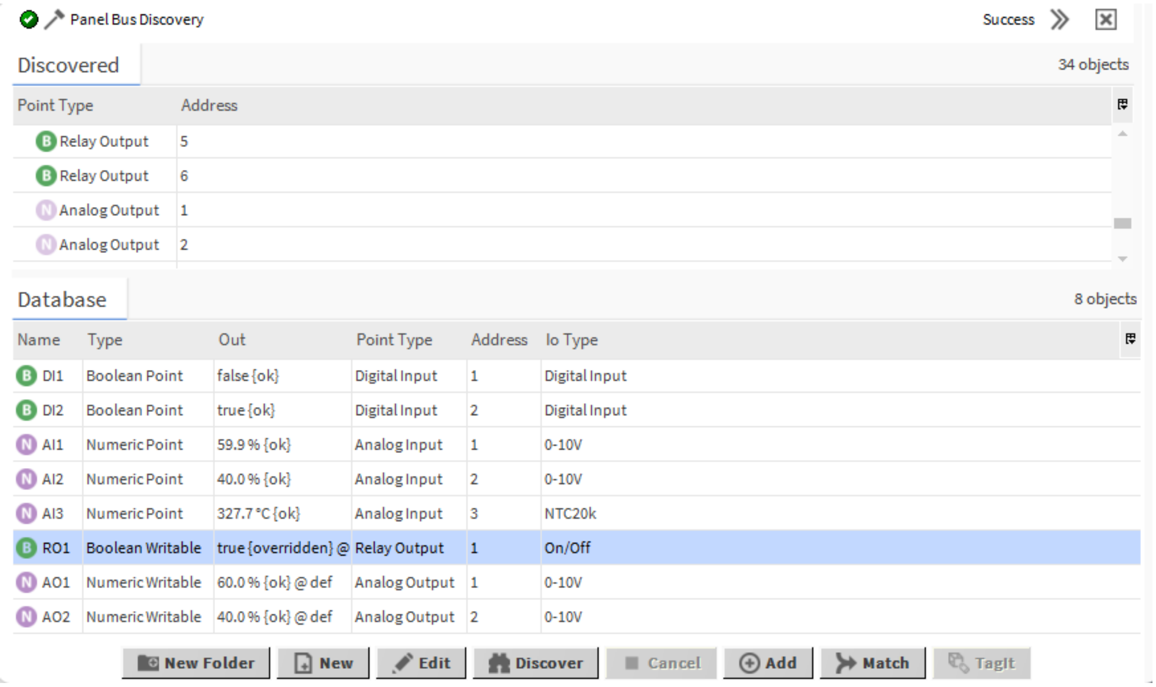
Analog Input is mapped into *Numeric Point* by default, but it can also be used as a digital input, then it is mapped into *Boolean Point*. **IO Type** allows to setup analog input type: select thermistor curve for passive sensors or 0/2-10V for active ones.

Digital Input is mapped into *Boolean Point* by default, but it can also be used as a counter input, then it is mapped into *Numeric Point*.

Analog Output is mapped into *Numeric Writable* by default, but it can also be used as a digital output, then it is mapped into *Boolean Writable*. **IO Type** allows to setup analog output type: 0-10V or 2-10V.

Relay Output is mapped into *Boolean Writable*.

Floating Output is mapped into *Numeric Writable*.



Discovered — 34 objects

Point Type	Address
B Relay Output	5
B Relay Output	6
N Analog Output	1
N Analog Output	2

Database — 8 objects

Name	Type	Out	Point Type	Address	Io Type
B DI1	Boolean Point	false {ok}	Digital Input	1	Digital Input
B DI2	Boolean Point	true {ok}	Digital Input	2	Digital Input
N AI1	Numeric Point	59.9 % {ok}	Analog Input	1	0-10V
N AI2	Numeric Point	40.0 % {ok}	Analog Input	2	0-10V
N AI3	Numeric Point	327.7 °C {ok}	Analog Input	3	NTC20k
B RO1	Boolean Writable	true {overridden} @	Relay Output	1	On/Off
N AO1	Numeric Writable	60.0 % {ok} @ def	Analog Output	1	0-10V
N AO2	Numeric Writable	40.0 % {ok} @ def	Analog Output	2	0-10V

Figure 3: Panel Bus points

Point units are set automatically in **Facets** property. All temperatures are in degrees Celsius by default. For Fahrenheit degrees set **Facets** to Fahrenheit and **Proxy Ext / Device Facets** to Celsius – values will be recalculated automatically.

Proxy Ext / Tuning Policy Name assigns *Tuning Policy* to each point. This policy determines when the point shall be written to, if the point is always subscribed and if the open / short circuit of analog inputs shall be mapped into point fault status.

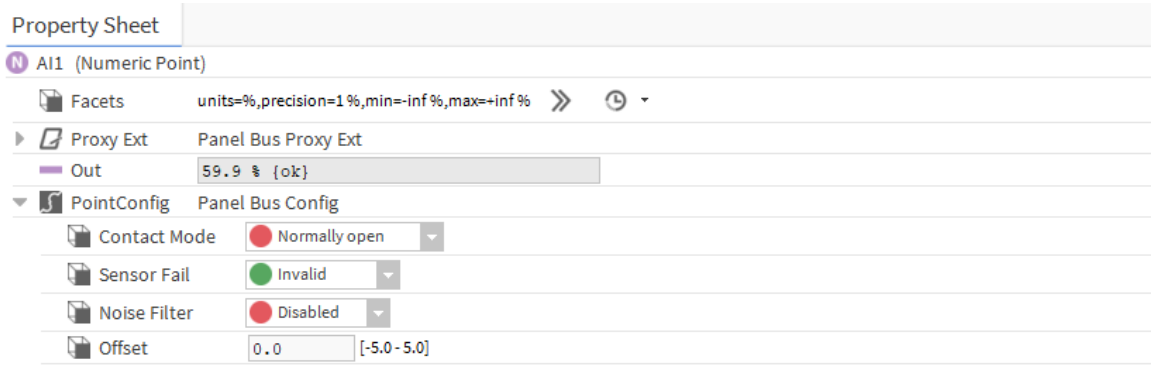
All added points are polled by the driver in groups to minimize the traffic. Polling rate is specified in **Panel Bus Device / Poll Frequency** property.

7 Panel Bus Point Config

Panel Bus points have extra configuration parameters, which determine for example whether DI is normally open or close and in which position shall RO remain in case of communication failure. If order to set these parameters **PointConfig** component shall be copied from panelBus palette under each point. **PointConfig** automatically displays parameters relevant to parent point type, see below. Driver checks if this component is available and writes its values into the Panel Bus module.

7.1 Digital Inputs

- **Contact Mode** – *Normally closed* or *Normally open*
- **LED Behavior** – IO LED colour to be *Off / Yellow* or *Green / Red* (except for CLIOP830A / CLIOP831A modules)



Property Sheet

AI1 (Numeric Point)

Facets	units=%,precision=1 %,min=-inf %,max=+inf %
Proxy Ext	Panel Bus Proxy Ext
Out	59.9 % {ok}
PointConfig	Panel Bus Config
Contact Mode	Normally open
Sensor Fail	Invalid
Noise Filter	Disabled
Offset	0.0 [-5.0 - 5.0]

Figure 4: Panel Bus point config

7.2 Analog Inputs

- **Contact Mode** – [for digital input mode] *Normally closed* or *Normally open*
- **Sensor Fail** – in case of sensor failure show XXX (*Invalid* option) or the last good value (*Last Position* option)
- **Noise Filter** – limits signal changes to filter out extra noise
- **Offset** – an offset to correct the value. One can also use **ProxyExt / Conversion** property to add offset on Niagara side

7.3 Digital and Analog Outputs

- **Direct Reverse** – allows to reverse output
- **Safety Position** – output signal in case of communication fault: 0%, 50%, 100% or keep the last value

7.4 Floating Outputs

- **Safety Position** – output signal in case of communication fault
- **End Switches** – if the motor has end switches, open and close relays will remain energized at the end positions
- **Power Up Sync** – the motor will fully close on power-up for synchronization
- **Sync 24h** – the motor will fully close every 24 hours for synchronization
- **Sync Break** – if *No* option is selected, complete synchronization before moving to a new position.
- **Close Repeat** – after reaching 0%, the close relay will be energized for a short period to fully close its rubber seal
- **Valve Exercising** – open valve once a week to assure reliable operation
- **Open Runtime** – time from 0% to 100%
- **Close Runtime** – time from 100% to 0%
- **Sync Charge** – how long the relay shall be kept energized after reaching end position during synchronization
- **Sync Level Open** – synchronize the motor to fully open position if its setpoint is higher than this setting; if the value is 255, disable this function
- **Sync Level Close** – synchronize the motor to fully closed position if its setpoint is lower than this setting; if the value is 255, disable this function
- **Min Stop Time** – minimal stop time before changing movement direction
- **Min Run Percent** – minimal percentage change