



Getting Started Guide

Pickering Sequence Manager



© Copyright (2021) Pickering Interfaces. All Rights Reserved.

No part of this publication may be reproduced, transmitted, transcribed, translated or stored in any form, or by any means without the written permission of Pickering Interfaces.

Technical details contained within this publication are subject to change without notice.



ISO 9001

Reg No.
FM38792

Introduction

The Pickering Sequence Manager software can be found on our downloads page:

<http://www.downloads.pickeringtest.info/downloads/>

Running the executable will prompt you to accept a license agreement, as well as select a number of installation options.

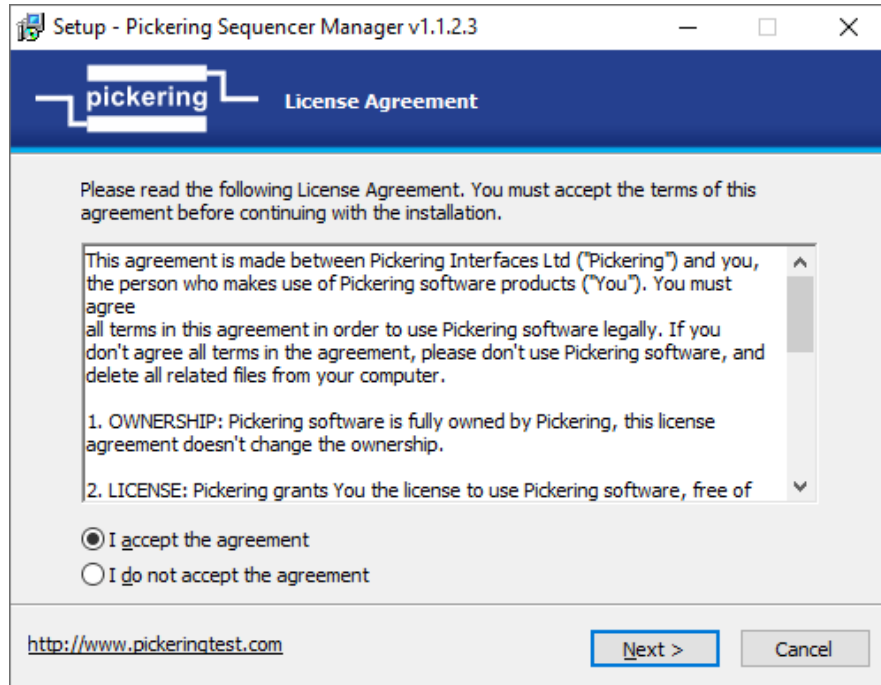


Figure 1. License Agreement Window

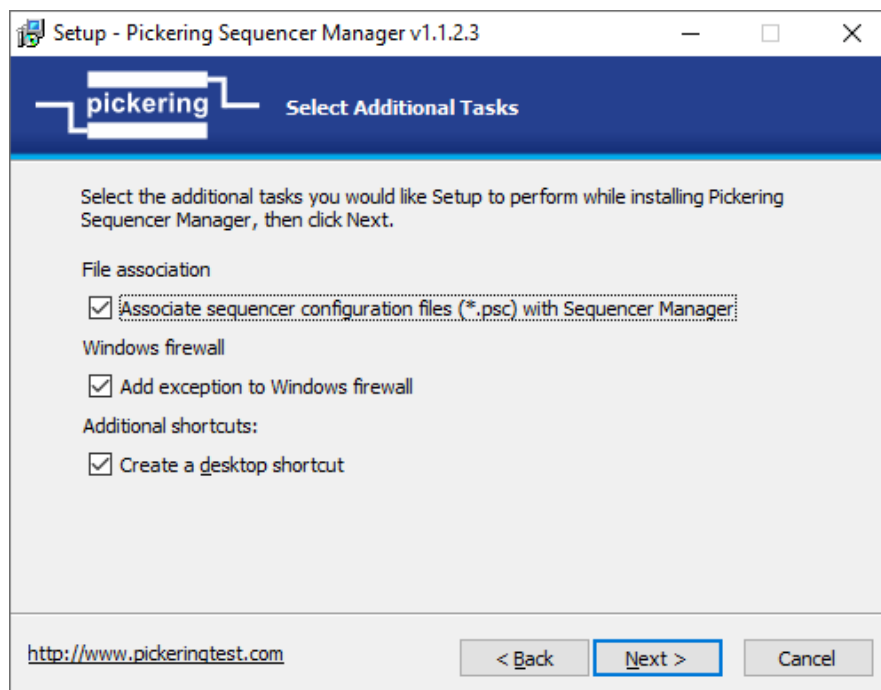


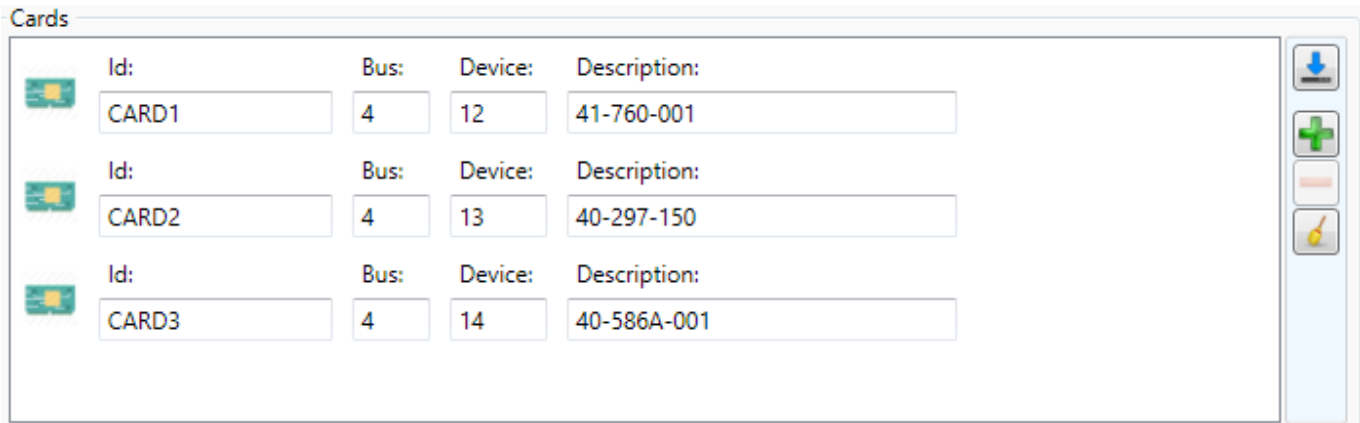


Figure 2. Installation Options

Creating a New Configuration and Adding Modules

A new configuration can be created by selecting "New" under the "Configuration" menu in the top left of the application. From here, modules can be added by selecting the  "Add Card" icon on the right of the screen. They can also be automatically loaded from a connected LXI chassis with the  "Import cards from connected device" icon. When a module is added this way the Id, Bus, Device, and Description fields are populated based on the modules discovered.






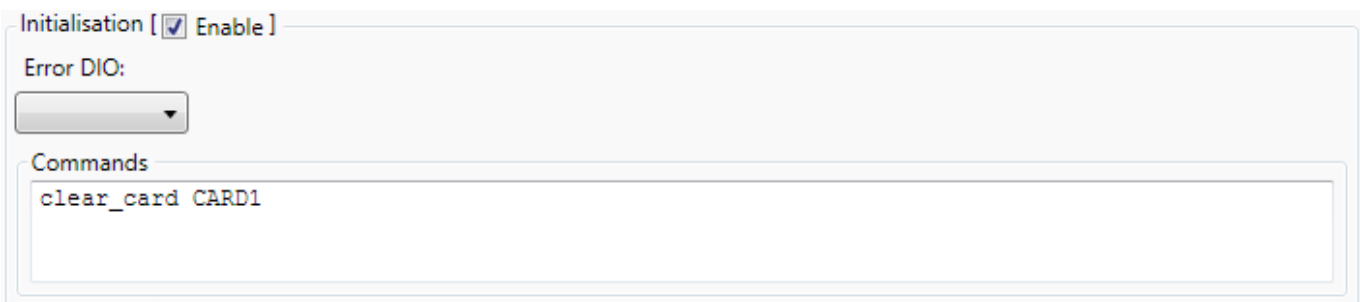
	Id:	Bus:	Device:	Description:
	CARD1	4	12	41-760-001
	CARD2	4	13	40-297-150
	CARD3	4	14	40-586A-001

Figure 3. Card List

Initialization Options

After cards are defined, initialization options can be configured. Below the Card and Subunit information, a check box for Enable will allow commands to be entered in a text box. These commands will be run when the sequence service is started and before any triggers are applied. These can be useful for ensuring a device state before any new connections are made.



Initialisation [Enable]

Error DIO:


Commands

```
clear_card CARD1
```

Figure 4. Command Text Box For Card Initialization

Defining Triggers

The second tab of the main window lists user-defined inputs and outputs, as well as the properties of the trigger signal. The trigger in and out timing can be specified along with whether to use a hardware trigger, software trigger, or both. If a trigger is defined as an input trigger the In-timer value (in microseconds) is the minimum length of the pulse monitored by the sequencing service to register it as a valid. The value in Out-timer defines the length of the pulse generated by the output triggers. The value can be used between 1-255 microseconds and is common to all triggers with the same direction. The node option defines the software process handling the trigger signals when using not using the exclusive hardware mode.

The IO controls can be added using the  "Add IOct!" button on the right of the window. Each of these IO lines can be assigned an identifier and direction. Software IO entries will also have the option to select a Signal while Hardware IO entries will require specifying a hardware pin and whether the trigger should occur on the Falling or Rising Edge.

In-timer (us)	Out-timer (us)	Mode	Node
40	60	Exclusive software mode	/dev/piltrig0

Id	Direction	DIO	Falling edge	Rising edge	Signal
TRG	In	Dio1 - Pin 2	<input checked="" type="radio"/>	<input type="radio"/>	User defined signal #1
ACK	Out	Dio2 - Pin 3	<input type="radio"/>	<input checked="" type="radio"/>	
DNF	Out	Dio3 - Pin 4	<input type="radio"/>	<input checked="" type="radio"/>	
ERR	Out	SwError - Pin 14	<input checked="" type="radio"/>	<input type="radio"/>	User defined signal #2

Figure 5. Inputs & Outputs Window

Creating Sequences

The third tab of the main window is used to define the Sequences associated with each trigger.

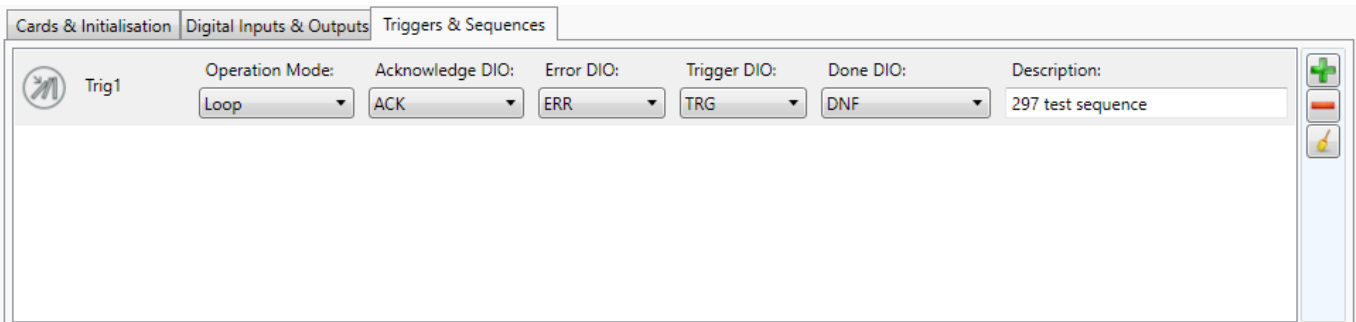




Figure 6. Trigger List

A new trigger can be added using the  "Add Trigger" button to the right of the trigger list. Each trigger will have options for the Operation Mode, Acknowledge DIO, Error DIO, Trigger DIO, Done DIO, and a field for a description. There are several operation modes available:

- **Wait Mode:** When all sequences are completed the chassis will enter an idle state. No external triggers will affect the chassis until the device receives a Restart or Reload command.
- **Loop Mode:** When all sequences are completed the chassis will wait for a trigger signal before starting again at the first sequence.
- **Unload Mode:** When all sequences are issued, the current configuration is unloaded. No external triggers will affect the chassis until a Reload command is received which will load the current configuration and listen for a trigger to begin sequencing again at the first sequence.

The Acknowledge DIO, Error DIO, Trigger DIO, and Done DIO entries will each have the option to use one of the Inputs or Outputs defined previously or left blank if that signal is not required.

Once a trigger is defined, sequences can be assigned to it. Select the  "Add sequence" button to create an entry in the sequence list. Each sequence has options to enable Acknowledge signals, closure limit checking, and whether errors are ignored. Additionally, the Order number provided will set the order in which the sequences are run. The "One sequence per trigger" option determines whether the full sequence list of a given trigger is run with a single received trigger signal or each sequence requires its own received trigger to run.

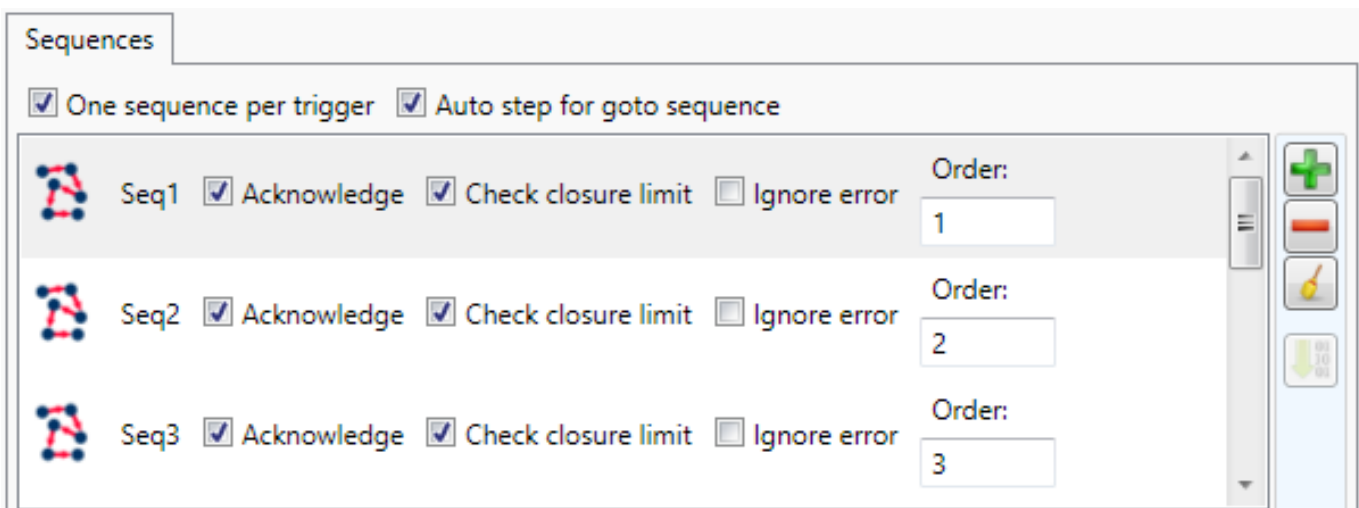



Figure 7. Sequence List

Adding Commands

The Commands field allows entry of the list of commands to be performed when a given sequence is run. Clicking the  "Show command list window" button will bring up a window of available commands as well as examples of their usage. These commands are also described in this document.

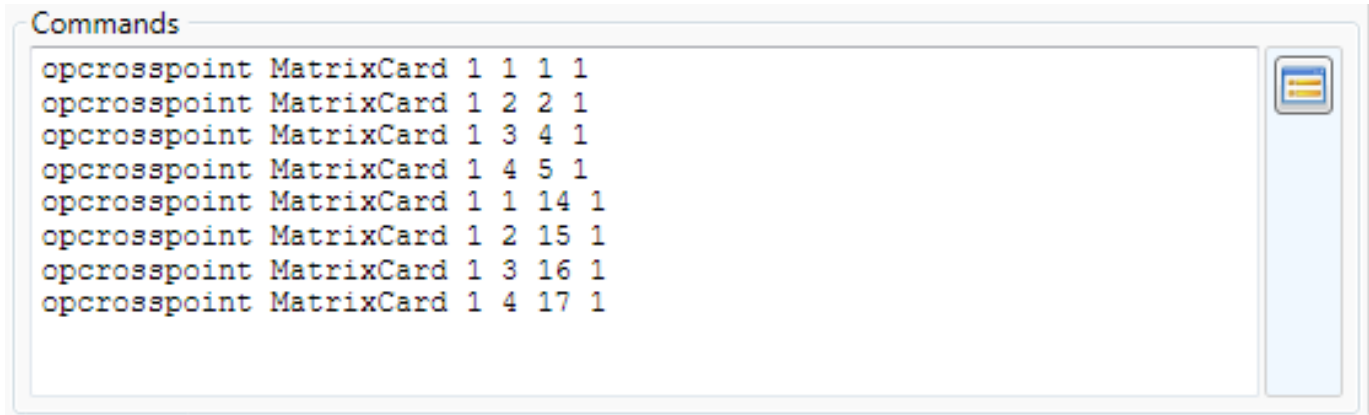


Figure 8. Command List Window

A red outline will appear around the command list when there is a syntax error:

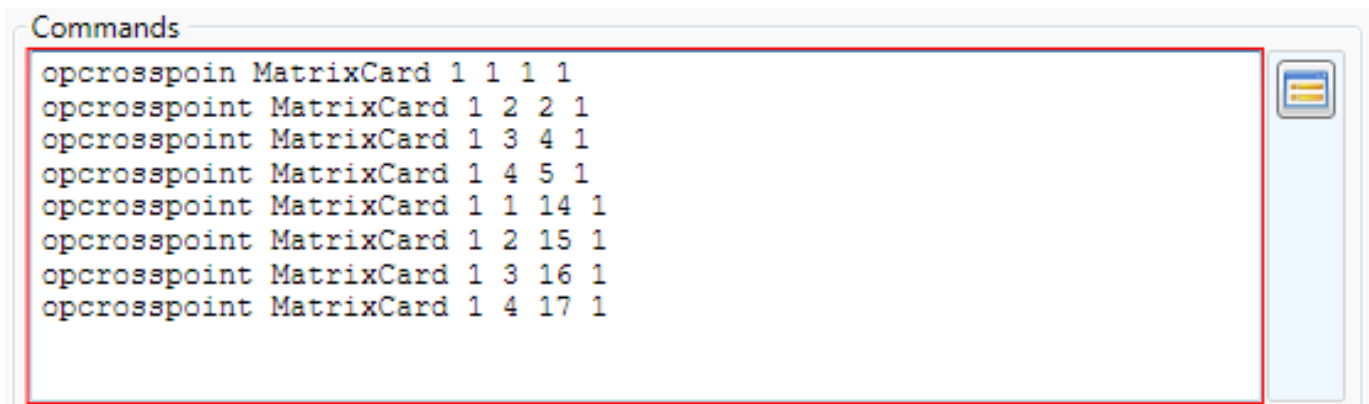




Figure 9. Command List With Syntax Error

Connecting to a Chassis

The generated sequences can be run from Sequence Manager after connecting to a compatible LXI chassis. The application can automatically discover networked devices or they can be manually added. Automatic discovery can be initiated using the  “Discover new devices” button after selecting a network adapter from the drop-down menu. A device can be manually added using the  “Add device to list manually” button.

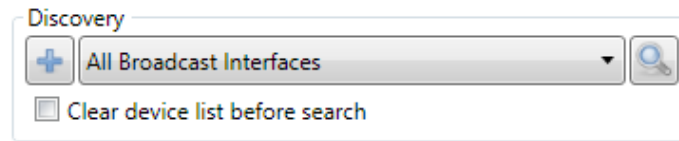




Figure 10. LXI Device Discovery Control



Either will populate the device list with the discovered LXI devices, regardless of sequencing compatibility. Non-compatible devices will appear in grey and a description of the highlighted device will appear below. Double-clicking an entry on the list will establish a connection to that device, and a status indicator is provided to show the current connection state.


Status	Device	Serial Number
	60-100-001 (Software Simulator)	LS0080027A4DAEA
	65-200-002	L380063
	60-555-027	L380158
	60-590-001 (60-901-001)	L380056
	60-590-002	L310016
	60-105-001	L380082




Description	LXI v1.4 Release
Firmware	5.2.4-rc4
IP Address	192.168.2.65
Model	60-105-001
Name	P105-L380082
Serial Number	L380082
Version	1.5 LXI Device Specification 2016

Figure 11. List of Discovered LXI Devices

Upload, Download, and Running Sequences

The  "Download configuration" and  "Upload configuration" options become available along the top center of the window once a connection is made to an LXI device. Downloading a configuration will retrieve the configuration currently on the device and load it in the current window, which will overwrite any unsaved data. Uploading a configuration will export the open configuration to the device but does not begin any sequences or enable triggering.

The  "Start sequence" option will set the device to listen for triggers. Once a trigger is provided the first sequence will be performed, followed by each subsequent sequence operating as defined previously with the "One sequence per trigger" and "Operation mode" options.

While the sequence is running, the  "Stop sequence",  "Go to sequence", and  "Restart sequence" options become available and can be used to control the flow of sequences run. Restarting the sequence will place it back in the initial state of waiting for a trigger to begin running from the first sequence.

Software triggers can be sent using the Pilmon tool which is accessed through an SSH interface to the LXI device. The command to send a trigger signal is:

```
STS    SendTriggerSignal    <signal(uint)> [<ack signal(uint)>]
      Sends a trigger signal (1 or 2) to sequence and wait, 2s timeout,
      for ack signal (1 or 2) if specified.
```

For more information on Pilmon or accessing the device using SSH, refer to our KnowledgeBase articles on these topics:

[SSH Control of LXI Devices](#)

[Pilmon Command Set](#)

Command List

General rules:

- Each command is on its own line
- Anything after “#” is treated as comment
- Sequence of tabs and spaces is considered as single white space character
- All numbers are decimal unless stated otherwise

Types used in sequencing commands:

- **dword** - unsigned 32-bit integer (corresponds to uint32_t C-type) It is default type used if not specified, for example: “sub” meaning Sub-Unit number. It is specified in decimal base unless noted otherwise.
- **bool** - unsigned 8-bit integer (corresponds to uint8_t C-type). Only two values are allowed:
0 - as FALSE
1 - as TRUE
- **float** - floating point number (corresponds to float C-type)
- **double** - floating point number (corresponds to double C-type)

NOTE: both ‘bool’ and ‘dword’ types have naming inherited from Windows™ API types even while Triggers and their ClientBridge library are running on Linux®.

Common labels:

- card - string - card alias, for example “card1”
- sub - dword, decimal - subunit number, for example “1”
- row - dword, decimal - matrix row
- col - dword, decimal - matrix column

Supported sequencing commands:

- **Clearing whole card - PIPLX_ClearCard()**
#clear_card card
clear_card card1
- **Clearing Sub-Unit of card - PIPLX_ClearSub()**
#clear_sub card
sub clear_sub card1 2
- **Write array of data to Sub-Unit - PIPLX_WriteSub()**
#write_sub card sub hex_dword1 hex_dword2
write_sub card2 1 00000000 00000000
- **Operate a single output channel or bit - PIPLX_OpBit()**
#opbit card sub bit_number action _bool
opbit card1 123 1
- **Operate a single matrix crosspoint - PIPLX_OpCrosspoint()**
#opcrosspoint card sub row column action _bool
opcrosspoint card1 1 1 1 1
- **Write to resistors - PIPLX_ResSetResistance()**
#set_res card sub res_value_double|short|open
set_rest card3 2 123.34
set_rest card3 2 short
set_rest card3 2 open

- **Set voltage source - PIPLX_VsourceSetVoltage()**
#vsource_setvoltage card sub voltage_double
vsource_setvoltage card1 1.23
- **Set attenuation value - PIPLX_AttenSetAttenuation()**
#set_atten card sub atten_float
set_atten card1 1.23
- **Sets the output voltage of battery simulator (BATT type) sub-units - PIPLX_BattSetVoltage()**
#batt_setvoltage card sub voltage_double
batt_setvoltage card1 1 1.23
- **Sets the output sink current of battery simulator (BATT type) sub-units - PIPLX_BattSetCurrent()**
#batt_setcurrent card sub current_amps_double
batt_setcurrent card1 1 0.12
- **Sets the output enable pattern of battery simulator (BATT type) sub-units - PIPLX_BattSetEnable()**
#batt_setenable card sub pattern_hex
batt_setenable card1 1 1f
- **Sets the value of a specific attribute - PIPLX_SetAttribute()**
#set_attribute card sub out_bool attr_code_hex_dword attr_value_hex_dword
#ATTR_MODE RESTRICTED
set_attribute card1 1 1 401 1
- **Sets the output voltage of a power supply sub-unit to the specified value - PIPLX_PsuSetVoltage()**
#psu_setvoltage card sub voltage_double
psu_setvoltage card1 1 1.23
- **Enables or disables a power supply's output - PIPLX_PsuEnable()**
#psu_enable card sub state_bool
psu_enable card1 1 1
- **Set mode - PIPLX_SetMode()**
#set_mode hex_number_of_mode
set_mode 1 # NO_WAIT
set_mode 0 # DEFAULT
- **Force error - simulate LXI error (useful for testing and debugging)**
#error name can be found as 'enum LXIErrorCodes' in Piplx.h (ClientBridge header)
#force_error name_of_lxi_error
force_error ER_CARD_DISABLED
- **Sleep for specified amount of seconds**
#sleep seconds
#sleep 2

LXI Chassis Relay Reset Function

Overview

Pickering's LXI Chassis Triggered Reset function provides a hardware interface, which upon a logic level change, will interrupt the LXI chassis controller and force the relays inside the chassis to their de-energized state. While this does provide a level of safety in the event of a system fault condition, it does not guarantee that all relays are in their de-energized state in the event that a relay has been welded shut.

The hardware interface is on an AUX port (25-pin Micro D-SUB male connector) located on the chassis controller. The input to the controller which can trigger a Reset event is the SW_RESET (pin 1). A separate output monitor pin, SW_ERROR (pin 14), will change state when a Reset condition has been detected.

Once an error condition has been detected, it will prevent further access to the chassis until the logic state on pin 1 has returned to Inactive.

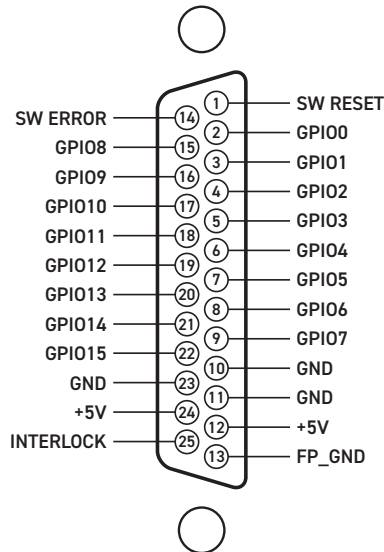


Figure 12. AUX Port Pinout (25-pin Micro D-SUB male connector)

AUX Port Signal Descriptions

Signal	Type	Output Stage (see images)	Max Sink/Source Current	Max Voltage	Function
GPIO(0-15)	I/O	Open Drain	Source 600µA (4.7kΩ pull-up), Sink 24mA	5V	General purpose IO for trigger functions.
SW RESET	I	CMOS Input Buffer With 10kΩ Pull-down	—	5V	Input signal to command software reset, active high.
SW ERROR	O	Open Drain	Source 600µA (4.7kΩ pull-up), Sink 24mA	5V	SW error output, programmable rising / falling edge. Logic high indicates a reset condition has occurred and chassis is in 'failsafe' mode.
GND	O	—	—	—	System power supply, ground reference.
+5V	O	—	1A (current limited)	—	System power supply, +5V.
INTERLOCK	I	CMOS Input With 4.7kΩ Pull-down	—	5V	Interlock for disabling module, active high. Connect to provided +5V pin for correct operation (module enabled).
FP_GND	O	—	—	—	Metalwork reference point, to be used for screen connection of interconnection accessories.

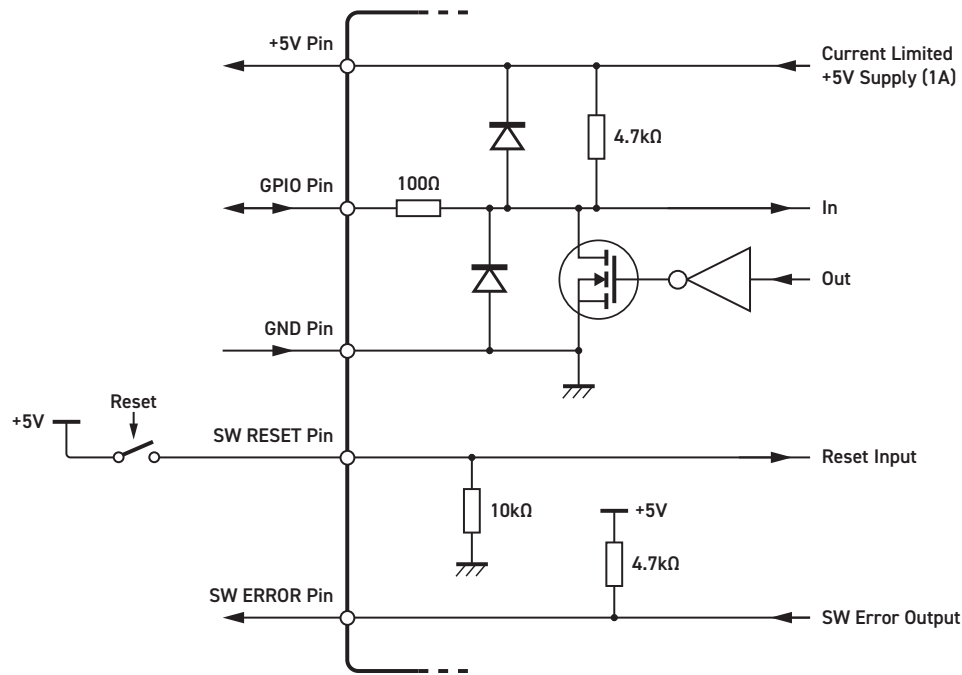


Figure 13. AUX Port Internal Circuitry

Applicable Hardware Information:

1. The Triggered Reset capability is available on all Pickering LXI chassis with model numbers that end in -002.
2. Minimum Firmware required is revision 5.9.1
3. 25-position Micro D-SUB connector (ITT Canon MDSM-25SC-Z11-VS1)



Figure 14. 25-pin Micro D-SUB Connector Components

4. SW_RESET (Input)
 - a) Rising or falling edge (programmable)
 - b) Minimum pulse width: 100 ms (programmable to 10 s)
 - c) Logic level: TTL (2.4 - 5V)
5. SW_ERROR (Output)
 - a) Active level: 5V, indicates a reset condition has occurred and chassis is in 'failsafe' mode.

Enabling the Triggered Reset Function

The Diagnostics page (Figure 15) must be accessed to enable the Triggered Reset function. By default, the function is disabled.

To activate the function

1. Select "Enable" from the Feature Control dropdown menu
2. Select the desired edge of the input signal on Pin 1 that will cause the Reset function to occur
3. The default minimum pulse width required to cause a Reset event is 100 ms. This can be changed to a value up to 10 seconds.
4. Select the 'Clear Reset Verification" method to enabled. This will require the SW_RESET line to be inactive before the Reset status can be cleared and communications to the box restored.
5. Click on the Submit button to save all changes.

Triggering Reset/Error ▼

Reset line settings:

Warning: Changing any settings here will destroy all active ServerBridge sessions.

Feature control

Pulse slope

Pulse minimum width [ms]

Clear Reset Verification

Reset/Error line status:

Line	Status
Reset	Not active
Error	Not active

Figure 15. LXI Chassis Diagnostics Page

Notes:

1. The Diagnostics Page displays the current status of the Reset and Error lines.
2. The SW_ERROR line is activated only when a Reset condition has occurred.
3. Changing settings can only occur when the Reset Flag and SW_ERROR output is inactive and will interrupt communication to all modules. All modules will be reset and relays will change to their de-energized state.

Application Example (testing the functionality with 1-step verification)

Detecting a Reset condition on a **Rising** edge (Pin 1), min pulse width is 100 ms.

1. From the *Diagnostics* page, view the options for **Triggering Reset/Error**
2. Select "Enable", "Rising" edge and, "100" for minimum pulse width.
3. Select "Disabled" from the Clear Reset Verification drop-down menu
4. Click on "Submit" – The box will be interrupted and all relays will return to their de-energized state.
5. Ensure in the system log that you see following line:
 - a) `<some date/time> (none) daemon.info ServerBridge: Triggering Reset/Error handler started (3013977184). (SW Reset Handler)`
6. Connect the hardware (+) signal that will be used to drive a Reset event to pin 1 (SW_RESET) and the (-) signal to pin 11 (GND)
7. Using application code or Pickering's soft front panel, close relays on the switch modules in the LXI chassis and manually verify that the selected relays have energized
8. Drive a rising edge TTL signal on pin 1 (SW_RESET) allowing the signal to remain logic Hi. This will lock the chassis and all pilpxi modules will be reset. Manually confirm that the relays closed in step 7 are now in their de-energized state. Also confirm that SW_ERROR output, pin 14, is Active (logic Hi).
9. The following message should be displayed in the message log
 - a) `<some date/time> (none) daemon.notice ServerBridge: Trigger type pulse signal change is stable. Reset line activated on rising edge. Resetting all pilpxi based cards.. (SW Reset Handler)`
 - b) `<some date/time> (none) daemon.info ServerBridge: All pilpxi based cards are protected, only read-only mode allowed. Waiting for clear signal to disable protection.. (SW Reset Handler)`
10. Confirm that no commands to operate relays can be completed by attempting to energize relays using application code or the Pickering supplied Soft Front Panel
11. Return the signal on SW_RESET pin 1 to logic Lo.
12. Confirm the SW_ERROR pin 14 has returned to logic Lo.
13. Once cleared, the status should be reflected on the web page in the section Reset/Error line status:.
14. The following messages should also be displayed in the system log:
 - a) `<some date/time> (none) daemon.info ServerBridge: Clear signal received. Disabling protection, releasing all cards.. (SW Reset Handler)`
15. Confirm that the chassis will accept commands to operate the relays using either application software or the Pickering Soft Front Panel

Application Example (testing the functionality with 2-step verification)

Two-step verification may be useful as an extra means of security. In this mode, an additional step is required to return the box to normal operating mode.

Detecting a Reset condition on a **Rising** edge (Pin 1), min pulse width is 100 ms.

1. From the Diagnostics page, view the options for Triggering Reset/Error
2. Select "Enable", "Rising" edge and, "100" for minimum pulse width.
3. Select "Enable" from the Clear Reset Verification drop-down menu
4. Click on "Submit" – The box will be interrupted and all relays will return to their de-energized state.
5. Ensure in the system log that you see following line:
 - a) `<some date/time> (none) daemon.info ServerBridge: Triggering Reset/Error handler started (3013977184). (SW Reset Handler)`
6. Connect the hardware (+) signal that will be used to drive a Reset event to pin 1 (SW_RESET) and the (-) signal to pin 11 (GND)
7. Using application code or Pickering's soft front panel, close relays on the switch modules in the LXI chassis and manually verify that the selected relays have energized
8. Drive a rising edge TTL signal on pin 1 (SW_RESET) allowing the signal to remain logic Hi. This will lock the chassis and all pilpxi modules will be reset. Manually confirm that the relays closed in step 5 are now in their de-energized state. Also confirm that SW_ERROR output, pin 14, is Active (logic Hi).
9. The following message should be displayed in the message log:
 - a) `<some date/time> (none) daemon.notice ServerBridge: Trigger type pulse signal change is stable. Reset line activated on rising edge. Resetting all pilpxi based cards.. (SW Reset Handler)`
 - b) `<some date/time> (none) daemon.info ServerBridge: All pilpxi based cards are protected, only read-only mode allowed. Waiting for clear signal to disable protection.. (SW Reset Handler)`
10. Confirm that no commands to operate relays can be completed by attempting to energize relays using application code or the Pickering supplied Soft Front Panel
11. Return the signal on SW_RESET pin 1 to logic Lo.
12. Confirm the SW_ERROR pin 14 has not changed and remains latched at logic Hi.
13. Confirm that no commands to operate relays can be completed and a system error is returned when attempting to energize relays using application code or the Pickering supplied Soft Front Panel
14. To clear the lock/reset state, perform one of the following steps. But first, ensure that the input to SW_RESET on pin 1 is still inactive.
 - i) Manually:
From the Diagnostics Triggering Reset/Error menu, click on the Clear button
 - ii) Programmatically:
Invoke the function call from the Pickering ClientBridge API. (in case of C++ ClientBridge:
`PICMLX_ClearSWReset(SESSION SID);`
15. Once cleared, the status should be reflected on the web page in the section Reset/Error line status:.
16. The following messages should also be displayed in the system log:
 - a) `<some date/time> (none) daemon.info ServerBridge: Clear signal received. Disabling protection, releasing all cards.. (SW Reset Handler)`
17. Confirm that SW_ERROR pin 14 has been driven Lo and the chassis will accept commands to operate the relays using either application software or the Pickering Soft Front Panel.

The oscilloscope screen shot in figure 16 displays the moment in time where a Reset condition is activated on a falling edge (100ms). The yellow trace represents the SW_RESET (pin 1) and the green trace represents SW_ERROR (pin 14). All relays are de-energized when the SW_ERROR (Green trace) changes to the active Hi state which is after the minimum pulse width has been satisfied.

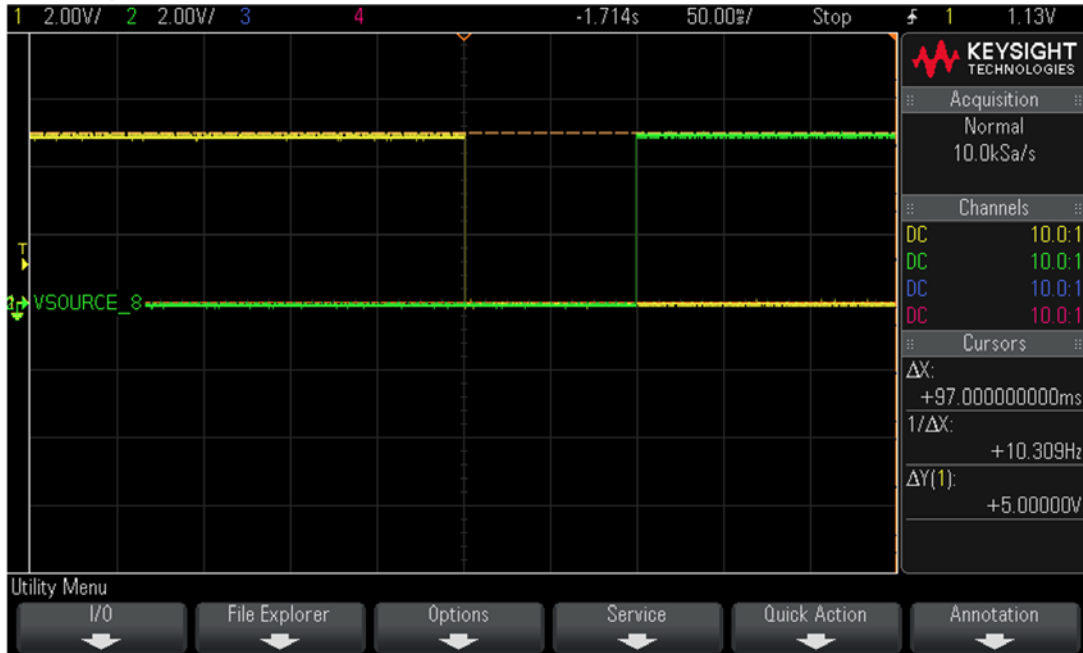


Figure 16. SW_ERROR and SW_RESET State Change Waveforms

Power Cycling

If the unit has experienced a failsafe condition, and power is cycled to the unit, as long as the input to SW_RESET returns to inactive before power is applied, the failsafe condition will clear and communication will be restored to the box. This is true regardless of the verification method. If SW_RESET remains active through the power cycle event, the failsafe condition will persist, SW_ERROR will remain active and the box will block commands to energize the relays.

Technical Support

For Technical Support please contact Pickering Interfaces either by phone, the website or via e-mail.

Worldwide Technical Support and Product Information:

pickeringtest.com

Pickering Interfaces Headquarters

Stephenson Road, Clacton-on-Sea, CO15 4NL United Kingdom

Tel: +44 (0)1255-687900

E-Mail: sales@pickeringtest.com

USA

Tel: (West) +1 541 471 0700

Tel: (East) +1 781 897 1710

E-mail: ussales@pickeringtest.com

France

Tel +33 9 72 58 77 00

E-mail: frsales@pickeringtest.com

Germany

Tel: +49 89 125 953 160

E-mail: desales@pickeringtest.com

Sweden

Tel: +46 340-69 06 69

E-mail: ndsales@pickeringtest.com

Czech Republic

Tel: +420 558 987 613

E-mail: desales@pickeringtest.com

China

Tel: +86 4008 799 765

E-mail: chinasales@pickeringtest.com