

---

# **OpenDCS Documentation**

***Release 0.4***

**Geoff Johnson**

November 07, 2016



<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Sample Layout . . . . .	3
1.2	Existing Features . . . . .	3
1.3	Planned Features . . . . .	4
1.4	Other Things that are Present but Non-functional . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Requirements . . . . .	5
2.2	Building from Source . . . . .	5
<b>3</b>	<b>Configuration</b>	<b>9</b>
3.1	Analog Input Control . . . . .	9
3.2	Analog Output Control . . . . .	9
3.3	Box . . . . .	10
3.4	Channel Treeview . . . . .	11
3.5	Channel Tree Entry . . . . .	12
3.6	Channel Tree Category . . . . .	13
3.7	Command Execution Control . . . . .	13
3.8	Log Control . . . . .	13
3.9	Page . . . . .	14
3.10	PID Control . . . . .	14
3.11	Plugin Control . . . . .	15
3.12	PNID Control . . . . .	15
3.13	PnidElement . . . . .	16
3.14	Chart . . . . .	16
3.15	Chart Axis . . . . .	17
3.16	Real Time Chart . . . . .	18
3.17	Real Time Chart Trace . . . . .	18
3.18	Data Series . . . . .	21
3.19	Real Time Multi-Channel Chart Trace . . . . .	21
3.20	Stripchart . . . . .	22
3.21	Stripchart Trace . . . . .	24
3.22	Polar Chart . . . . .	24
3.23	Polar Chart Axis . . . . .	25
3.24	Color Map . . . . .	26
3.25	Heat Map . . . . .	27
3.26	Polar Heat Map . . . . .	28
3.27	Channel Matrix . . . . .	29

3.28	Channel Matrix Element . . . . .	30
<b>4</b>	<b>Usage</b>	<b>31</b>
4.1	Page Selection . . . . .	31
4.2	Title Block . . . . .	31
4.3	Setting Page . . . . .	31
4.4	Configuration . . . . .	31
<b>5</b>	<b>Indices and tables</b>	<b>33</b>

Contents:



## Introduction

Dactl is an application for creating custom data acquisition and control systems under the GNOME desktop environment. At this time it is heavily dependent on the existence of a valid configuration, the details of which are entirely undocumented.

There is a companion library that dactl is pretty heavily reliant on and will be referenced to through the documentation, [libcdl](#). It has it's own (incomplete) documentation which can be viewed [here](#).

This software is still in early stages of development

### 1.1 Sample Layout

Using [this](#) configuration file dactl will generate the view seen here.

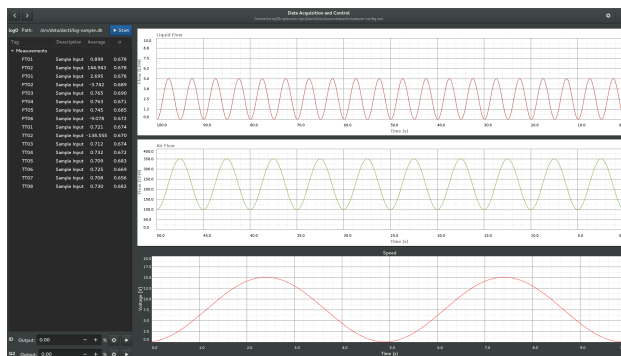


Fig. 1.1: A sample dactl layout configuration.

### 1.2 Existing Features

- XML configurable UI classes for:
  - AI channel
  - AO channel
  - DI channel
  - DO channel

- Log file (start/stop)
  - Strip chart
- UI for changing the properties of:
  - libcld channel types (AI/AO/DI/DO/Math)
  - libcld data series (incomplete)
  - libcld channel calibrations
  - libcld logs
- A page to export logged database tables as CSV files
- Framework for custom plugins. See the *example plugin*. <<https://github.com/coanda/dactl-mcc-plugin>>

## 1.3 Planned Features

- Configuration-less mode
- DBus API for adding and configuring UI
- XML configurable UI classes for:
  - Recording video from OpenCV compatible cameras
  - Chart for viewing recorded history of measurement data
  - Data browser that connects to stored CSV and SQLite logs
- UI for changing the properties of:
  - General application settings
  - All dactl UI widgets
- Peas plugin loader
- Ability to tweak dactl UI widgets using Gtk properties of underlying classes

## 1.4 Other Things that are Present but Non-functional

- All plugins in src/plugins other than Velmex which is only partially complete
- A page to edit the configuration file using gtksourceview



---

## Installation

---

### 2.1 Requirements

- Linux is the only tested OS
- GNOME 3 is the only tested DE
- Vala

### 2.2 Building from Source

The source code is hosted on [GitHub](#).

#### 2.2.1 Pre-installation Setup

##### Install Fedora 19 .. 23 dependencies

```
1 sudo dnf install -y automake autoconf libtool gnome-common intltool gcc vala
2 sudo dnf install -y glib2-devel gtk3-devel libxml2-devel libgee-devel \
3   json-glib-devel clutter-devel clutter-gtk-devel gsl-devel gtksourceview3-devel \
4   libmatheval-devel sqlite-devel gobject-introspection-devel gettext-devel \
5   gettext-common-devel libmodbus-devel comedilib-devel librsvg2-devel \
6   python3-devel pygobject3-devel
```

##### Install Ubuntu 14.04 dependencies

Notice: These commands have only been tested as part of a Travis-CI build.

```
1 sudo add-apt-repository ppa:vala-team/ppa -y
2 sudo apt-get update -qq
3 sudo apt-get install -qq gnome-common libglib2.0-dev libjson-glib-dev \
4   libgee-0.8-dev libvala-0.22-dev libgsl0-dev libsqlite0-dev libxml2-dev \
5   libmatheval-dev libmodbus-dev libcomedi-dev valac-0.22 librsvg2-dev \
6   libgirepository1.0-dev libgtk-3-dev libclutter-1.0-dev libclutter-gtk-1.0-dev \
7   python3-dev python-gobject-dev
```

## 2.2.2 Compiled Dependencies

### Install Vala dependencies

```
1 git clone https://github.com/geoffjay/modbus-vapi.git
2 git clone https://github.com/geoffjay/comedi-vapi.git
3 sudo mkdir -p /usr/local/lib/pkgconfig
4 sudo cp comedi-vapi/comedi.pc /usr/local/lib/pkgconfig/
5 ver=`vala --version | sed -e 's/.*\([0-9]\.[0-9][0-9]\).*\1/'`
6 sudo cp comedi-vapi/comedi.vapi /usr/share/vala-$ver/vapi/
7 sudo cp modbus-vapi/libmodbus.vapi /usr/share/vala-$ver/vapi/
```

### Install libclld

```
1 git clone https://github.com/geoffjay/libclld.git
2 cd libclld
3 git checkout v0.3.1
4 export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
5 ./autogen.sh
6 make && sudo make install
7 cd ..
8 echo "/usr/local/lib" | sudo tee --append /etc/ld.so.conf
9 sudo ldconfig
```

## 2.2.3 Compile and Install dactl

**Warning:** Installation overwrites the configuration file at  $\$(sysconfdir)/dactl/$ , if an alternate value wasn't provided for `-prefix` than this is probably `/usr/local/etc/dactl`. It's recommended that the existing configuration is copied over `data/config/dactl.xml` or backed up and dealt with separately.

```
1 git clone https://github.com/coanda/dactl.git
2 cd dactl
3 export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
4 ./autogen.sh
5 make && sudo make install
```

## 2.2.4 Post-installation Configuration

The make install command given previously will overwrite the site-wide configuration, to fix the ownership settings you may need to do something along the lines:

```
1 chown -R `whoami`.$(id -gn `whoami`) /usr/local/etc/dactl
2 chmod -R g+w /usr/local/etc/dactl
3 chmod +x /usr/local/share/applications/dactl.desktop
```

## 2.2.5 Optional but Useful

Currently the only drivers tested for data acquisition hardware are comedi. You might be able to do something in dactl without comedi, but probably not. Some distributions (Ubuntu?) have support for comedi built into the kernel provided, but not Fedora. The instructions that we use for compiling comedi using dkms are

```
1 su -
2 dnf install -y automake autoconf libtool git dkms kernel-devel kernel-headers
3 git clone git://comedi.org/git/comedi/comedi.git
4 cp -R comedi/ /usr/src/comedi-0.7.76+20120626git-1.nodist
5 cd /usr/src/
6 dkms add -m comedi -v 0.7.76+20120626git-1.nodist
7 cd comedi-0.7.76+20120626git-1.nodist && ./autogen.sh && cd ..
8 dkms build -m comedi -v 0.7.76+20120626git-1.nodist
9 dkms install -m comedi -v 0.7.76+20120626git-1.nodist
10 echo "KERNEL==\"comedi*\", MODE=\"0666\", GROUP=\"iocard\" > /etc/udev/rules.d/95-comedi.rules
```

After these steps if you have a comedi compatible device you should be able to *modprobe comedi* as well as that for the device and it should show up in */dev*. If not, a test device can be created by:

```
1 su -
2 dnf install -y comedilib comedilib-devel
3 modprobe comedi comedi_num_legacy_minors=4
4 modprobe comedi_test
5 comedi_config /dev/comedi0 comedi_test
```

However, test devices are of limited use, they allow for instructions only on with no support <sup>1</sup> for commands.

---

<sup>1</sup> At least not that I'm aware of.



---

## Configuration

---

A collection of user interface elements are available. Configuration refers here to the XML code that defines the individual elements. Included in the collection are the box and page and thus a layout is created by recursively packing elements into boxes which are in turn packed into pages.

### 3.1 Analog Input Control

This control displays the current value along with statistical data.

This section shows how to add an analog input control to a configuration file.

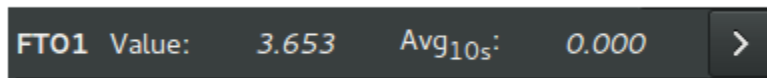
```
<ui:object id="ai-ctl0" type="ai" ref="/daqctl0/dev0/ai00"/>
```

#### Table of Configurable Attributes

attribute	default value
id	ai-ctl0
ref	null

This class contains no configurable properties.

The Analog Input Control as it appears when added to the interface:



### 3.2 Analog Output Control

This control facilitates

```
<ui:object id="ao-ctl0" type="ao" ref="/daqctl0/dev0/ao00"/>
```

#### Table of Configurable Attributes

attribute	default value
id	ao-ctl0
ref	null

This class contains no configurable properties.

## 3.3 Box

This section shows how to add a box to a configuration file. A Dactl Box inherits properties of a Gtk Box. It is simply a container for a user interface element or another box.

```
1 <ui:object id="box0" type="box">
2   <ui:property name="homogeneous">true</ui:property>
3   <ui:property name="orientation">horizontal</ui:property>
4   <ui:property name="expand">true</ui:property>
5   <ui:property name="fill">true</ui:property>
6   <ui:property name="spacing">0</ui:property>
7   <ui:property name="margin-top">0</ui:property>
8   <ui:property name="margin-right">0</ui:property>
9   <ui:property name="margin-bottom">0</ui:property>
10  <ui:property name="margin-left">0</ui:property>
11  <ui:property name="hexpand">true</ui:property>
12  <ui:property name="vexpand">true</ui:property>
13  <!--
14    - Can contain <ui:object> elements of type:
15    -
16    - * "ai":          Dactl.AIControl
17    - * "ao":          Dactl.AOControl
18    - * "box":         Dactl.Box
19    - * "tree":        Dactl.ChannelTreeView
20    - * "chart":       Dactl.Chart
21    - * "stripchart":  Dactl.StripChart
22    - * "rt-chart":    Dactl.RTChart
23    - * "polarchart":  Dactl.PolarChart
24    - * "pnid":        Dactl.Pnid
25    - * "pid":         Dactl.PidControl
26    - * "exec":        Dactl.ExecControl
27    - * "log":         Dactl.LogControl
28    - * "video":       Dactl.VideoProcessor
29    -->
30 </ui:object>
```

#### Table of Configurable Attributes

attribute	default value
id	null
name	null

#### Table of Configurable Properties

property	data type	default value
homogeneous	bool	null
orientation	string	null
expand	bool	null
fill	bool	null
spacing	int	null
margin-top	int	null
margin-right	int	null
margin-bottom	int	null
margin-left	int	null
hexpand	bool	null
vexpand	bool	null

## 3.4 Channel Treeview

This section shows how to add a channel treeview to a configuration file. The columns of the treeview will appear in the same order as the configuration file. Channels can be put in to groups that can expand or collapse when the category name is activated.

```

1 <ui:object id="tree0" type="tree">
2   <ui:property name="width-request">width-re</ui:property>
3   <ui:property name="show-header">true</ui:property>
4   <ui:property name="expand">true</ui:property>
5   <ui:property name="fill">true</ui:property>
6   <ui:property name="show-tag">true</ui:property>
7   <ui:property name="show-desc">true</ui:property>
8   <ui:property name="show-sample-sdev">true</ui:property>
9   <ui:property name="show-sample-size">true</ui:property>
10  <ui:property name="show-units">true</ui:property>
11  <!--
12    - Can contain <ui:object> elements of type:
13    -
14    - * "tree-category": Dactl.ChannelTreeCategory
15    - * "tree-entry":   Dactl.ChannelTreeEntry
16    -->
17 </ui:object>

```

**Table of Configurable Attributes**

attribute	default value
id	null
chref	null

**Table of Configurable Properties**

property	data type	default value
width-request	int	null
show-header	bool	null
expand	bool	null
fill	bool	null
show-tag	bool	null
show-desc	bool	null
show-sample-sdev	bool	null
show-sample-size	bool	null
show-units	bool	null

The Channel Treeview Control as it appears when added to the interface:

Tag	Description	Average	$\sigma$
▼ Measurements			
SPX	Spare Input	3.679	0.010
FT06	Flow	-6.423	0.020
FT07	Flow	-6.409	0.018
FT08	Flow	-6.421	0.017
FT09	Flow	-6.386	0.015
FT10	Flow	-6.409	0.013
FT11	Flow	-6.368	0.009
FT12	Flow	-6.410	0.006
SPX	Spare Input	-6.368	0.008
SPX	Spare Input	-6.422	0.012
SPX	Spare Input	-6.371	0.008
TT01	Temperature	-6.437	0.014

## 3.5 Channel Tree Entry

This section shows how to add a channel tree entry to a configuration file.

```
<ui:object id="entry0" type="tree-entry" chref="/daqctl0/dev0/ai00"/>
```

**Table of Configurable Attributes**

attribute	default value
id	null
chref	null



## 3.6 Channel Tree Category

This section shows how to add a channel tree category to a configuration file.

```

1 <ui:object id="cat0" type="tree-category"/>
2   <ui:property name="title">Title</ui:property>
3   <!--
4     - Can contain <ui:object> elements of type:
5     -
6     - * "tree-category": Dactl.ChannelTreeCategory
7     - * "tree-entry":    Dactl.ChannelTreeEntry
8     -->
9 </ui:object>

```

**Table of Configurable Attributes**

attribute	default value
id	null

**Table of Configurable Properties**

property	data type	default value
title	string	null

## 3.7 Command Execution Control

This section shows how to add a control that executes the given command to a configuration file.

```

1 <ui:object id="exec-ctl0" type="exec"/>

```

**Table of Configurable Attributes**

attribute	default value
id	null

This class contains no configurable properties.

## 3.8 Log Control

This section shows how to add a log control to a configuration file.

```

1 <ui:object id="log-ctl0" type="log" ref="/logctl0/log0"/>

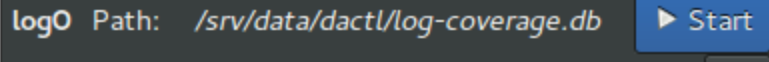
```

**Table of Configurable Attributes**

attribute	default value
id	null
ref	null

This class contains no configurable properties.

The Log Control as it appears when added to the interface:



### 3.9 Page

This section shows how to add a page to a configuration file.

```

1 <ui:object id="pg0" type="page">
2   <ui:property name="index">0</ui:property>
3   <ui:property name="title">Title</ui:property>
4   <ui:property name="expand">true</ui:property>
5   <ui:property name="fill">true</ui:property>
6   <ui:property name="visible">true</ui:property>
7   <!--
8     - Can contain <ui:object> elements of type:
9     -
10    - * "box": Dactl.Box
11    -->
12 </ui:object>

```

#### Table of Configurable Attributes

attribute	default value
id	null

#### Table of Configurable Properties

property	data type	default value
index	int	null
title	string	null
expand	bool	null
fill	bool	null
visible	bool	null

### 3.10 PID Control

This section shows how to add a pid control to a configuration file.

```

1 <ui:object id="pid-ctl0" type="pid" ref="/atmctl0/pid0"/>

```

#### Table of Configurable Attributes

attribute	default value
id	null
ref	null

This class contains no configurable properties.

The PID Control as it appears when added to the interface:



## 3.11 Plugin Control

This section shows how to add a plugin control to a configuration file.

```

1 <ui:object id="plugin-ctl0" type="plugin-control" parent="box0">
2   <!-- Can contain references to the CLD tree, eg. -->
3   <ui:property name="ref">/daqctl/dev0/ao00</ui:property>
4 </ui:object>

```

### Table of Configurable Attributes

attribute	default value
null	null

### Table of Configurable Properties

property	data type	default value
ref	string	null

## 3.12 PNID Control

This section shows how to add a pnid to a configuration file.

```

1 <ui:object id="pnid0" type="pnid">
2   <ui:property name="image-file">image-file.svg</ui:property>
3   <ui:property name="expand">true</ui:property>
4   <ui:property name="fill">true</ui:property>
5   <ui:property name="timeout">1000</ui:property>
6   <!--
7     - Can contain <ui:object> elements of type:
8     -
9     - * "pnid-text": Dactl.PnidElement
10    -->
11 </ui:object>

```

### Table of Configurable Attributes

attribute	default value
id	null

### Table of Configurable Properties

property	data type	default value
image-file	string	null
expand	bool	null
fill	bool	null
timeout	int	null

## 3.13 PnidElement

This section shows how to add a PNID to a configuration file.

```
1 <ui:object id="element0" type="element">
2   <ui:property name="cld-ref">cld-ref</ui:property>
3   <ui:property name="svg-ref">svg-ref</ui:property>
4 </ui:object>
```

**Table of Configurable Attributes**

attribute	default value
id	null
cld-ref	null
svg-ref	null

**Table of Configurable Properties**

property	data type	default value
cld-ref	string	null
svg-ref	string	null

## 3.14 Chart

This section shows how to add a chart to a configuration file.

```
1 <ui:object id="chart0" type="chart">
2   <ui:property name="title">Title</ui:property>
3   <ui:property name="expand">true</ui:property>
4   <ui:property name="fill">true</ui:property>
5   <ui:property name="height-min">0</ui:property>
6   <ui:property name="weight-min">0</ui:property>
7   <ui:property name="show-title">true</ui:property>
8   <ui:property name="show-grid">true</ui:property>
9   <ui:property name="show-grid-border">true</ui:property>
10  <!--
11    - Can contain <ui:object> elements of type:
12    -
13    - * "chart-axis": Dactl.Axis
14    -->
15 </ui:object>
```

**Table of Configurable Attributes**

attribute	default value
id	null

**Table of Configurable Properties**

property	data type	default value
title	string	null
expand	bool	null
fill	bool	null
height-min	int	null
width-min	int	null
show-title	bool	null
show-grid	bool	null
show-grid-border	bool	null

## 3.15 Chart Axis

This section shows how to add a chart axis to a configuration file.

```

1 <ui:object id="ax0" type="chart-axis">
2   <ui:property name="label">true</ui:property>
3   <ui:property name="orientation">horizontal</ui:property>
4   <ui:property name="min">true</ui:property>
5   <ui:property name="max">true</ui:property>
6   <ui:property name="div-major">0</ui:property>
7   <ui:property name="div-minor">0</ui:property>
8   <ui:property name="show-label">true</ui:property>
9   <ui:property name="show-minor-ticks">true</ui:property>
10  <ui:property name="show-major-ticks">true</ui:property>
11  <ui:property name="show-minor-labels">true</ui:property>
12  <ui:property name="show-major-labels">true</ui:property>
13  <ui:property name="show-start-label">true</ui:property>
14  <ui:property name="show-end-label">true</ui:property>
15  <ui:property name="rotate-label">true</ui:property>
16 </ui:object>

```

**Table of Configurable Attributes**

attribute	default value
id	null

**Table of Configurable Properties**

property	data type	default value
label	bool	null
orientation	string	null
min	bool	null
max	bool	null
div-major	int	null
div-minor	int	null
show-labe	bool	null
show-minor-ticks	bool	null
show-major-ticks	bool	null
show-minor-labels	bool	null
show-major-labels	bool	null
show-start-label	bool	null
show-end-label	bool	null
rotate-label	bool	null

## 3.16 Real Time Chart

This section shows how to add a real time chart to a configuration file.

```

1 <ui:object id="chart2" type="rt-chart">
2   <ui:property name="title">Speed</ui:property>
3   <ui:property name="height-min">100</ui:property>
4   <ui:property name="width-min">100</ui:property>
5   <ui:property name="refresh-ms">33</ui:property>
6   <ui:property name="show-grid">true</ui:property>
7   <ui:property name="show-grid-border">true</ui:property>
8   <ui:property name="show-title">true</ui:property>
9   <ui:property name="reverse-x-axis">false</ui:property>
10  <ui:property name="show-x-axis-label">true</ui:property>
11  <ui:property name="rotate-x-axis-label">false</ui:property>
12  <ui:property name="show-y-axis-label">true</ui:property>
13  <ui:property name="rotate-y-axis-label">true</ui:property>
14  <!--
15    - Can contain <ui:object> elements of type:
16    -
17    - * "chart-axis": Dactl.Axis
18    - * "rt-chart-trace": Dactl.Trace
19    -->
20 </ui:object>

```

**Table of Configurable Attributes**

attribute	default value
id	null

**Table of Configurable Properties**

property	data type	default value
title	string	null
expand	bool	null
fill	bool	null
height-min	int	null
width-min	int	null
show-title	bool	null
show-grid	bool	null
show-grid-border	bool	null
refresh-ms	int	33

The real time chart as it appears when added to the interface:

Double clicking the chart reveals basic configuration options:

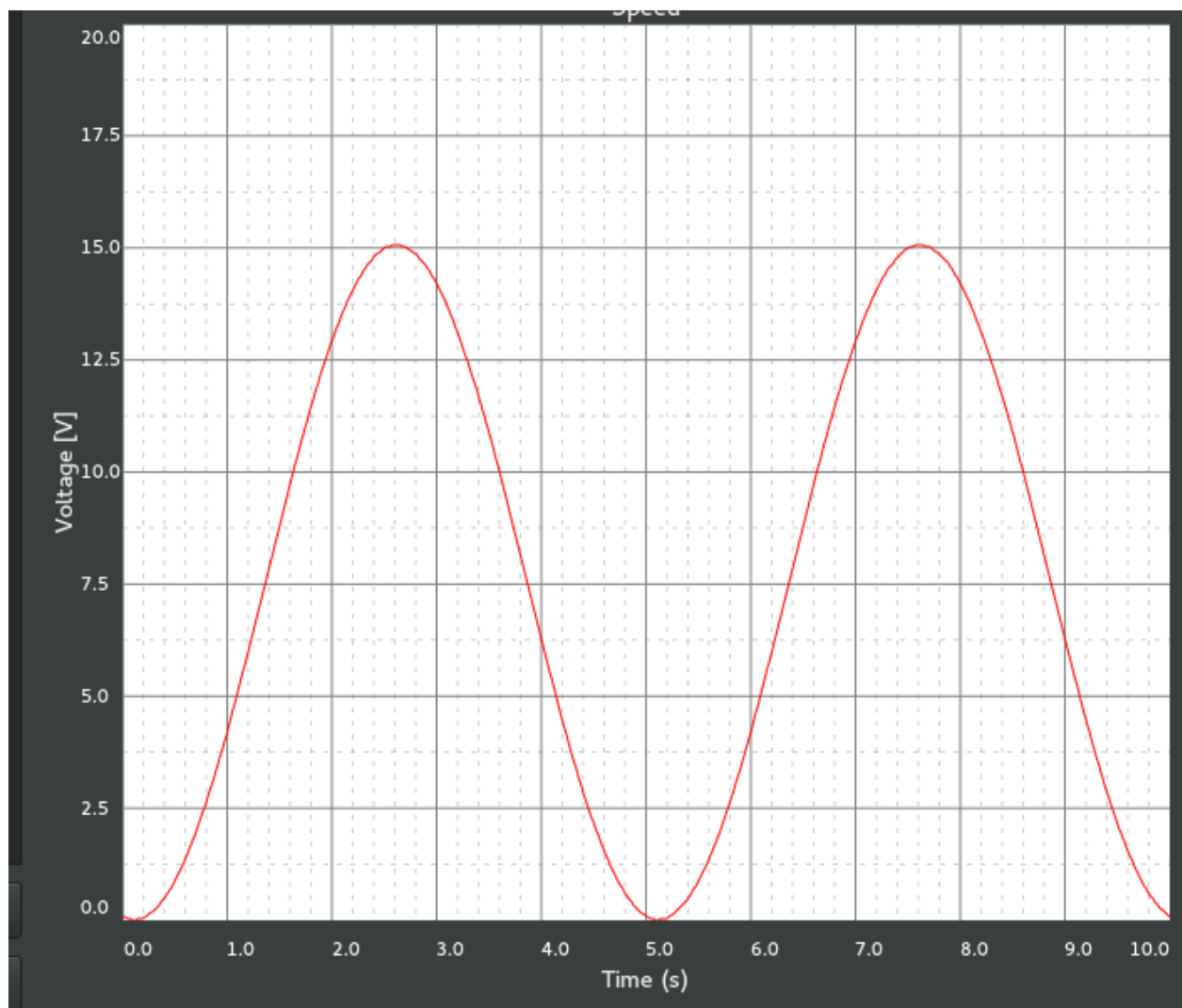
## 3.17 Real Time Chart Trace

This section shows how to add a real time chart trace to a configuration file.

```

1 <ui:object id="tr0-0" type="trace" ttype="real-time">
2   <ui:property name="color">rgb(255,0,0)</ui:property>
3   <ui:property name="line-weight">1</ui:property>
4   <ui:property name="draw-type">line</ui:property>
5   <ui:property name="points">1000</ui:property>
6   <!--

```



Y Axis:	Voltage [V]		
Min:	0.000	-	+
Max:	20.000	-	+
Major:	8	-	+
Minor:	2	-	+
X Axis:	Time (s)		
Min:	0.000	-	+
Max:	10.000	-	+
Major:	10	-	+
Minor:	5	-	+
Title:	Speed		
Traces			



```

7     - Can contain <ui:object> elements of type:
8     -
9     - * "dataseries": Dactl.DataSeries
10    -->
11 </ui:object>

```

#### Table of Configurable Attributes

attribute	default value
id	null

#### Table of Configurable Properties

property	data type	default value
points	int	null
draw-type	Dactl.TraceDrawType	null
line-weight	double	null
color	string	null

## 3.18 Data Series

A data series is a buffer that can be used to hold trace data.

This section shows how to add a data series to the configuration file.

```

1 <ui:object id="ds0-0" type="dataseries" ref="/daqctl0/dev0/ai02">
2   <ui:property name="buffer-size">1000</ui:property>
3   <ui:property name="stride">1</ui:property>
4 </ui:object>

```

#### Table of Configurable Attributes

attribute	default value
id	null
ref	null

#### Table of Configurable Properties

property	data type	default value
buffer-size	int	null
stride	int	null

## 3.19 Real Time Multi-Channel Chart Trace

This trace type can display data from several channels in a single trace.

This section shows how to add a real time chart trace to a configuration file.

```

1 <ui:object id="pglchart0tr0" type="trace" ttype="multichannel">
2   <ui:property name="color">#ce5c00</ui:property>
3   <ui:property name="line-weight">1</ui:property>
4   <ui:property name="draw-type">line</ui:property>
5   <!--
6     - Can contain <ui:object> elements of type:
7     -
8     - * "channel-vector": Dactl.ChannelVector

```

```
9      -->
10 </ui:object>
```

#### Table of Configurable Attributes

attribute	default value
id	null
ref	null

#### Table of Configurable Properties

property	data type	default value
draw-type	Dactl.TraceDrawType	null
line-weight	double	null
color	string	null

## 3.20 Stripchart

This is the legacy strip chart. The settings user interface for this was in need of improvement and so it was replaced by the real time chart. It has been retained in the library because it has some advantages over the newer chart type. Because its traces do not interpolate the data, they look much better with noisy data than the newer real time chart trace.

This section shows how to add a stripchart to a configuration file.

```
1 <ui:object id="chart0" type="stripchart">
2   <ui:property name="title">Title</ui:property>
3   <ui:property name="expand">true</ui:property>
4   <ui:property name="fill">true</ui:property>
5   <ui:property name="height-min">0</ui:property>
6   <ui:property name="width-min">0</ui:property>
7   <ui:property name="show-title">true</ui:property>
8   <ui:property name="show-grid">true</ui:property>
9   <ui:property name="show-grid-border">true</ui:property>
10  <ui:property name="points-per-second">10</ui:property>
11  <!--
12    - Can contain <ui:object> elements of type:
13    -
14    - * "chart-axis": Dactl.Axis
15    - * "stripchart-trace": Dactl.Trace
16    -->
17 </ui:object>
```

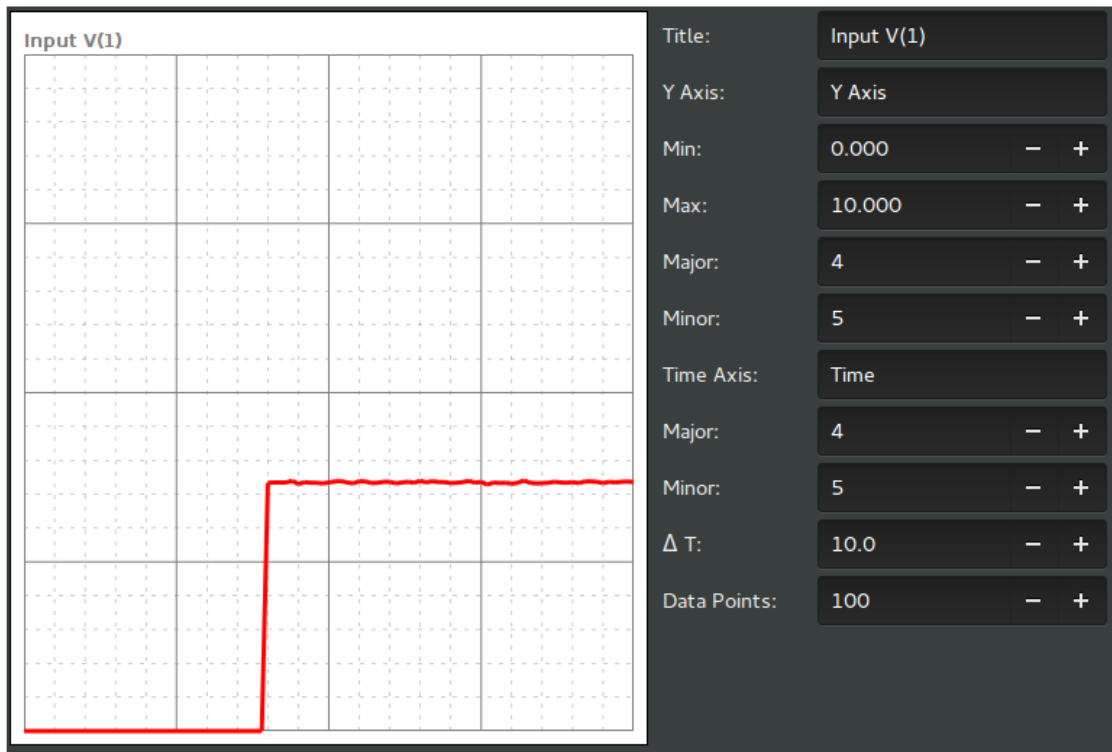
#### Table of Configurable Attributes

attribute	default value
id	null

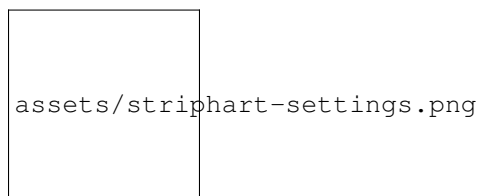
#### Table of Configurable Properties

property	data type	default value
title	string	null
expand	bool	null
fill	bool	null
height-min	int	null
width-min	int	null
show-title	bool	null
show-grid	bool	null
show-grid-border	bool	null
points-per-second	int	null

The Stripchart as it appears when added to the interface:



Double clicking the chart reveals basic configuraton options:



## 3.21 Stripchart Trace

This is the legacy stripchart trace that is used with the strip chart.

This section shows how to add a stripchart trace to a configuration file.

```

1 <ui:object id="tr0" type="stripchart-trace" ref="/daqctl0/dev0/ai00">
2   <ui:property name="buffer-size">100</ui:property>
3   <ui:property name="color">rgba(164,0,0,1.0)</ui:property>
4   <ui:property name="line-weight">1.0</ui:property>
5   <ui:property name="draw-type">line</ui:property>
6   <ui:property name="window-size">500</ui:property>
7   <ui:property name="stride">2</ui:property>
8 </ui:object>

```

**Table of Configurable Attributes**

attribute	default value
id	null
ref	null

**Table of Configurable Properties**

property	data type	default value
buffer-size	int	null
draw-type	Dactl.TraceDrawType	null
line-weight	double	null
color	string	null
stride	int	null
window-size	int	null
duration	string	null

## 3.22 Polar Chart

The polar chart is a surface for plotting polar data. To be complete it requires additional polar axes and a source of data. Currently, a heatmap is the only available drawable data source that can be displayed but trace data may be added in a future release.

```

1 <ui:object id="pglchart0" type="polar-chart">
2   <ui:property name="title">Bin Heat Map</ui:property>
3   <ui:property name="refresh-ms">30</ui:property>
4   <ui:property name="height-min">100</ui:property>
5   <ui:property name="width-min">100</ui:property>
6   <ui:property name="show-grid">true</ui:property>
7   <ui:property name="show-grid-border">true</ui:property>
8   <ui:property name="show-title">true</ui:property>
9   <ui:property name="zoom">0.9</ui:property>
10  <!--
11    - Can contain <ui:object> elements of type:
12    -
13    - * "chart-axis": Dactl.Axis
14    - * "colormap" Dactl.ColorMap
15    -->
16 </ui:object>

```

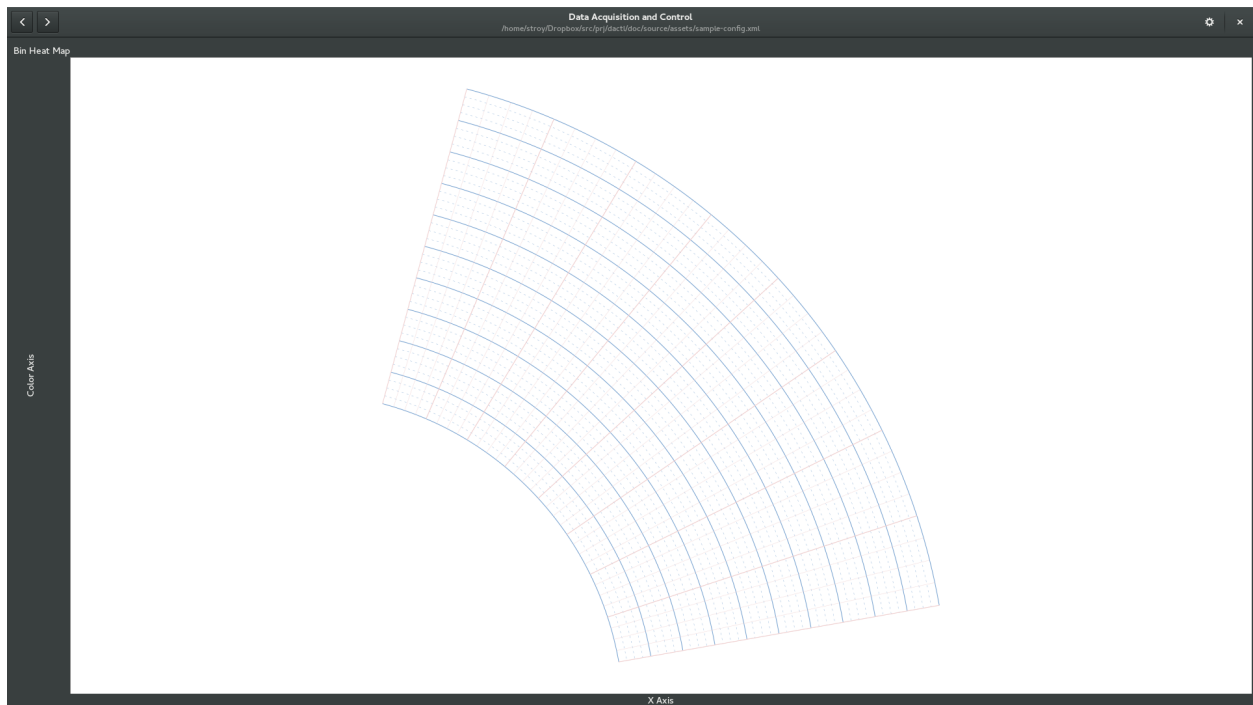
### Table of Configurable Attributes

attribute	default value
id	null

### Table of Configurable Properties

property	data type	default value
title	string	null
expand	bool	null
fill	bool	null
height-min	int	null
width-min	int	null
refresh-ms	int	33
zoom	double	0.8
show-title	bool	null
show-grid	bool	null
show-grid-border	bool	null

A polar chart with defined axes but no data.



## 3.23 Polar Chart Axis

This section shows how to add a polar chart axis to a configuration file.

```

1 <ui:object id="ax0" type="polar-chart-axis">
2   <ui:property name="label">Angle [deg]</ui:property>
3   <ui:property name="polar-axis-type">angle</ui:property>
4   <ui:property name="min">10</ui:property>
5   <ui:property name="max">75</ui:property>
6   <ui:property name="div-major">8</ui:property>

```

```

7  <ui:property name="div-minor">4</ui:property>
8  <ui:property name="color">rgb(240,206,206)</ui:property>
9  <ui:property name="show-major-ticks">true</ui:property>
10 <ui:property name="show-major-labels">true</ui:property>
11 <ui:property name="intersect-value">45</ui:property>
12 </ui:object>
13 <ui:object id="ax1" type="polar-chart-axis">
14   <ui:property name="label">Distance [in]</ui:property>
15   <ui:property name="polar-axis-type">magnitude</ui:property>
16   <ui:property name="min">5</ui:property>
17   <ui:property name="max">10</ui:property>
18   <ui:property name="div-major">10</ui:property>
19   <ui:property name="div-minor">4</ui:property>
20   <ui:property name="color">rgb(114,159,207)</ui:property>
21   <ui:property name="show-major-ticks">true</ui:property>
22   <ui:property name="show-major-labels">true</ui:property>
23   <ui:property name="intersect-value">10</ui:property>
24 </ui:object>

```

Table of Configurable Attributes

attribute	default value
id	null

Table of Configurable Properties

property	data type	default value
----------	-----------	---------------

## 3.24 Color Map

This element is added to the interface as a legend showing a gradient of colors between a set minimum and maximum color value. The gradient type is selected to interpolate colors as either RGB or HSV.

This section shows how to add a color map to a configuration file.

```

1  <ui:object id="cm0" type="colormap">
2    <ui:property name="min">0</ui:property>
3    <ui:property name="max">10</ui:property>
4    <ui:property name="div-major">10</ui:property>
5    <ui:property name="div-minor">4</ui:property>
6    <ui:property name="show-major-ticks">true</ui:property>
7    <ui:property name="show-minor-ticks">true</ui:property>
8    <ui:property name="show-major-labels">true</ui:property>
9    <ui:property name="min-color">rgba(0,0,255,0.8)</ui:property>
10   <ui:property name="max-color">rgba(255,0,0,0.8)</ui:property>
11   <ui:property name="gradient">rgb</ui:property>
12 </ui:object>

```

Table of Configurable Attributes

attribute	default value
id	null

Table of Configurable Properties

property	data type	default value
min   double   null		
max   double   null		
div-major	int	10
div-minor	int	2
show-minor-ticks	bool	true
show-major-ticks	bool	true
show-major-labels	bool	true
min-color	string	null
max-color	string	null
gradient	string	null

## 3.25 Heat Map

This section shows how to add a polar heat map to a configuration file.

```

1 <ui:object id="hmap-0" type="heatmap">
2   <ui:property name="xmin">0</ui:property>
3   <ui:property name="xmax">10</ui:property>
4   <ui:property name="ymin">0</ui:property>
5   <ui:property name="ymax">10</ui:property>
6   <ui:property name="zmin">0</ui:property>
7   <ui:property name="zmax">10</ui:property>
8   <ui:property name="min-color">rgba(114,159,207,0.8)</ui:property>
9   <ui:property name="max-color">rgba(239,41,41,0.8)</ui:property>
10  <ui:property name="interpolation-type">none</ui:property>
11  <ui:property name="rows">4</ui:property>
12  <ui:property name="columns">4</ui:property>
13  <!--
14    - Can contain <ui:object> element of type:
15    -
16    - * "channel-matrix": Dactl.ChannelMatrix
17    -->
18 </ui:object>

```

### Table of Configurable Attributes

attribute	default value
id	null

### Table of Configurable Properties

property	data type	default value
xmin   double   null		
xmax   double   null		
ymin   double   null		
ymax   double   null		
zmin   double   null		
zmax   double   null		
interpolation-type	string   "none"	
rows   int   null		
columns   int   null		

A chart with heatmap data as it appears when added to the interface.

Here the color values have been made translucent by editing the alpha value. This

allows the grid lines to show through.



## 3.26 Polar Heat Map

This section shows how to add a polar heat map to a configuration file.

```

1 <ui:object id="pg2chart0hm0" type="heatmap" subtype="polar">
2   <ui:property name="magnitude-min">0</ui:property>
3   <ui:property name="magnitude-max">10</ui:property>
4   <ui:property name="angle-min">0</ui:property>
5   <ui:property name="angle-max">360</ui:property>
6   <ui:property name="zmin">0</ui:property>
7   <ui:property name="zmax">10</ui:property>
8   <ui:property name="interpolation-type">none</ui:property>
9   <ui:property name="rings">10</ui:property>
10  <ui:property name="sectors">8</ui:property>
11  <!--
12    - Can contain <ui:object> element of type:
13    -
14    - * "channel-matrix": Dactl.ChannelMatrix
15    -->
16 </ui:object>

```

**Table of Configurable Attributes**

attribute	default value
id	null

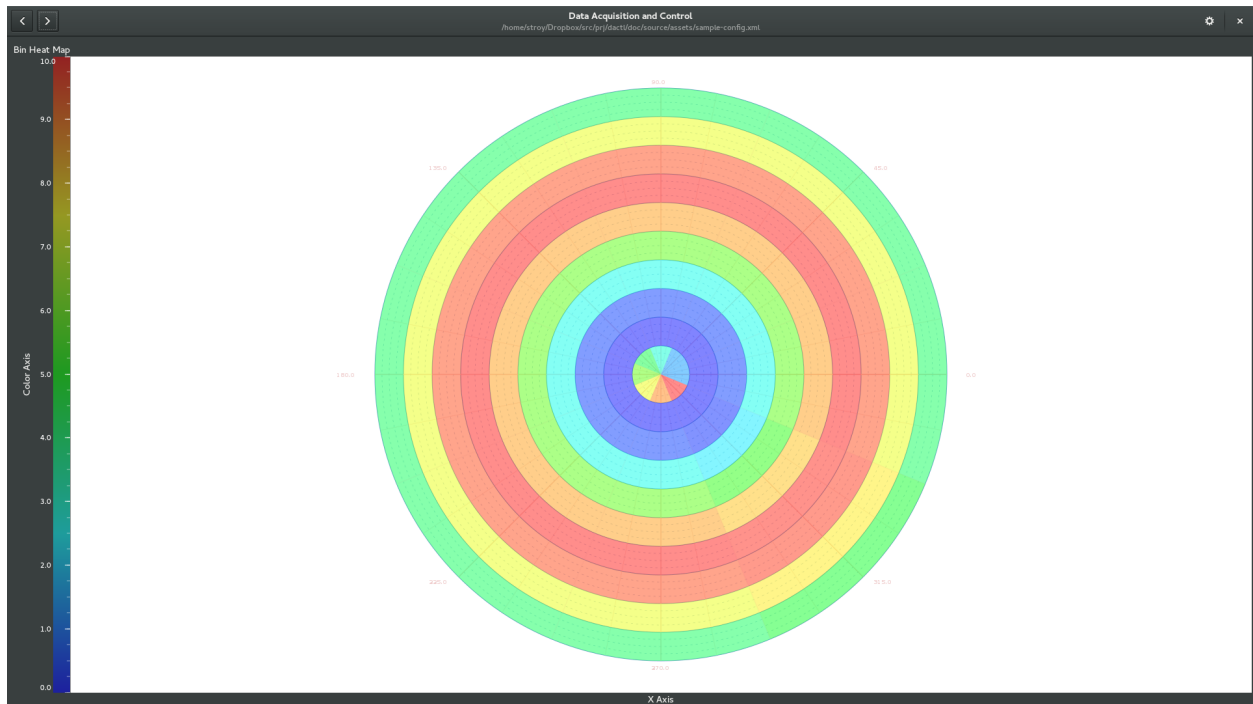
**Table of Configurable Properties**



property	data type	default value
magnitude-min	double	null
magnitude-max	double	null
angle-min	double	null
angle-max	double	null
interpolation-type	string	"none"
rings	int	null
sectors	int	null

A polar chart with polar heatmap data as it appears when added to the interface.

Here the color values have been made translucent by editing the alpha value. This allows the grid lines to show through.



## 3.27 Channel Matrix

A channel matrix is a container of triplet data points as required by the heatmap chart drawable types.

This section shows how to add a channel matrix to a configuration file.

```

1 <ui:object id="pg2chart0hm0ary0" type="channel-matrix">
2   <!--
3     - Can contain <ui:object> element of type:
4     -
5     - * "channel-matrix-element": Dactl.ChannelMatrixElement
6     -->
7 </ui:object>

```

Table of Configurable Attributes

attribute	default value
id	null

## 3.28 Channel Matrix Element

This section show how to add a channel matrix element to a configuration file as required by the channel matrix type.

```
1 <ui:object id="pg2chart0hm0ary0p00" type="channel-matrix-element">
2   <ui:property name="a">0.5</ui:property>
3   <ui:property name="b">22.5</ui:property>
4   <ui:property name="chref">/udp64</ui:property>
5 </ui:object>
```

### Table of Configurable Attributes

attribute	default value
id	null

### Table of Configurable Properties

property	data type	default value
a	double	null
b	double	null
chref	string	null

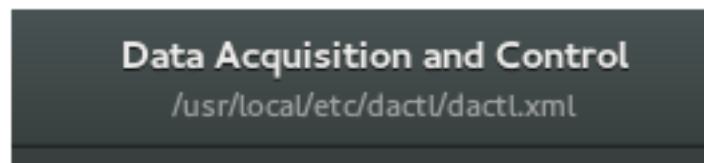
## 4.1 Page Selection

The following figure shows the page selection button:



## 4.2 Title Block

The following figure shows the title block:

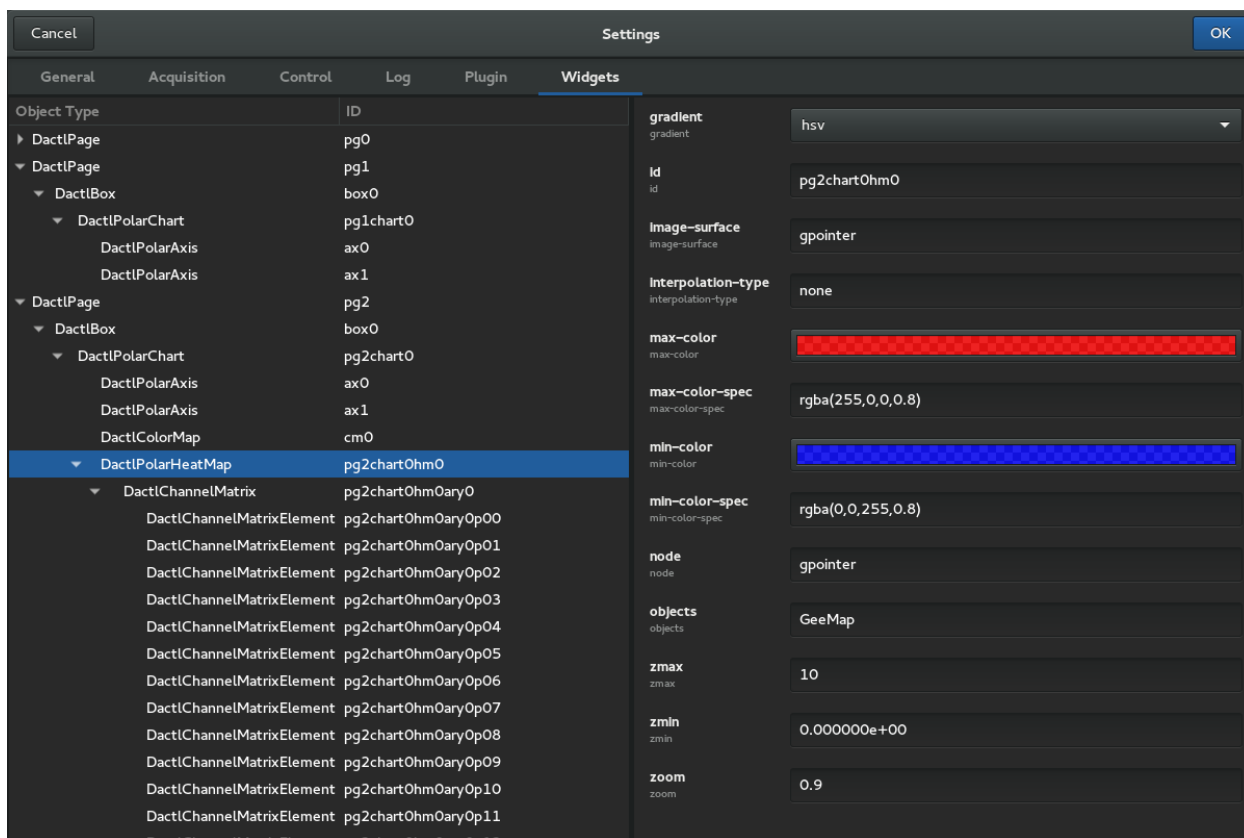
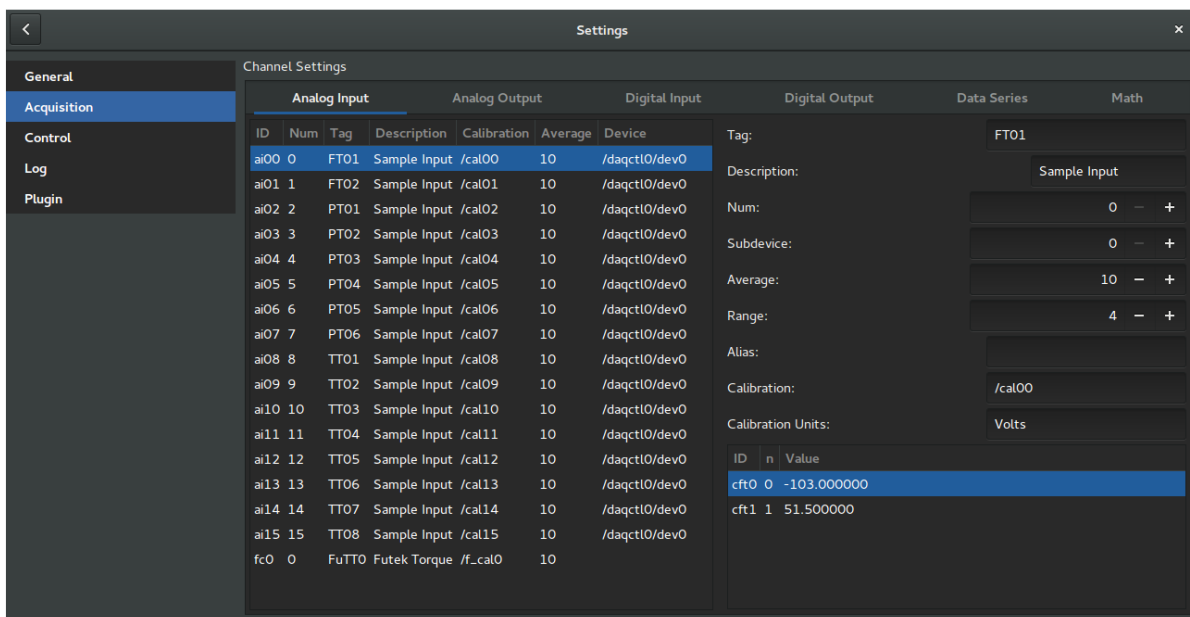


## 4.3 Setting Page

The following figure shows the setting page:

## 4.4 Configuration

The following figure shows the preferences button:



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`