Siglent Waterfall

Release 0.1

Eric Waller

Jan 03, 2020

CONTENTS

CHAPTER

ONE

INTRODUCTION

This program obtains trace information from a Siglent oscilloscope and displays them as a three dimension plot displaying a scrolling history of recent traces. The plot can be rotated and zoomed in our out using point devices.

The application is written in *Python3* on and uses multi-programming techniques to allow the program to use multiple cores.

Communication with the oscilloscope uses the *visa* instrument control infrastructure.

CHAPTER

TWO

USAGE

To start the program from the command line, see the following usage.:

```
Usage: siglent [OPTIONS]

  Siglent

  Display a YT display and (optionally) a waterfall display of scope traces
  obtained from an SDS 1204X-E oscilloscope.

Options:
  -r, --rows INTEGER       Rows in waterfall display (number of traces)
  -c, --cols INTEGER       columns in waterfall display (samples/trace)
  --waterfall              Show the 3D Display
  --channel TEXT           Scope channel to capture (default=C1)
  -n, --name TEXT          VISA Address of the oscilloscope  [required]
  -v, --verbose            Be verbose.  Invoke twice for more verbosity
  -L, --log-file PATH      Log file Name.
  -t, --timeout INTEGER    Foreground process timeout in seconds (debug only)
  -b, --bgtimeout INTEGER  Background process timeout in seconds (debug only)
  --help                   Show this message and exit.
```

For example:

```
siglent -n tcpip::192.168.1.201 -vv --waterfall
```

Will start the *Siglent Waterfall* program reading from the instrument at IP Address 192,168.1.201, being extremely verbose (debug level), and will display both the 2D and 3D displays.

```
siglent -n tcpip::192.168.1.201 -v
```

Will start the *Siglent Waterfall* program reading from the instrument at IP Address 192,168.1.201, being a little verbose (info level), and will display only 2D and 3D display.

CHAPTER

THREE

SIGLENTWATERFALL MODULES

3.1 Main

3.2 Background

Generate a stream of vectors read from a siglent oscilloscope

class background.**Acquire**(*address*, *channel*, *cols*)

Acquire traces from the oscilloscope

This class sets up the oscilloscope, then reads data from it continuously, posting those trace information to a queue that is processed by the main program. This code is run in a multiprocess environment allowing it to run independently of the main program. Hopefully, it runs on its own core.

Parameters

- **address** (*str*) – The *visa* address of the instrument
- **channel** (*str*) – The channel to read from the scope. Defaults to 'C1'. Values can be *C1, C2, C3, C4* or *math*
- **cols** (*int*) – The number of points to place in each vector. This controls the *skip factor* in the waveform setup. This decimates the 14K (or more) points in the scope memory down to a manaeable number of points

update()

Read the next trace from the oscillocope

Read parameters as to the format and scale of the trace, read the raw data points, apply scale and offset information to the data

Returns: *False* if there is an error. Otherwise, return a tuple of two equal size lists of float. The first element is a list of *float* representing time values, and a list of *float* representing voltages.

exception background.**ScopeInitializationError**

background.**main**(*cols*, *timeout*, *address*, *channel*, *pipe*)

Stream traces from a Siglent oscilloscope to a pipe.

Read traces from a Siglent 1204x-E oscilloscope and stream those traces to the main program through a pipe. We also will receive commands through that pipe. Handle those commands and respnd to those cmmands through the pipe as well. Commands and responses to those commands will be of the type 'Message' from the class defined above. The traces are sent as a 2-tuple of lists. The first list represents the time values of the samples, while the second list represents the voltages at the time values.

Parameters

- **cols** (*int*) – The expected size of the lists to be sent through the pipe.
- **timeout** (*int*) – The maximum time this process is to run. 0 implys no timeout
- **address** (*str*) – The *visa* address of the oscilloscope
- **channel** (*str*) – The name of the oscilloscope channel to stream.
- **pipe** (*pipe*) – The pipe through which data are communicated to and from the main program.

Returns *None*

Return type None

3.3 Log

log.**setLog**(*LogLevelStr*)

Set the log level to be used during this run. This program uses logging to provide warning, info, and debug level messages. Wwarnings are always enabled. info level messages are considered to be "Verbose".

Parameters **LogLevelStr** – A string representing on of the members of logging that define log levels ("CRITICAL", DEBUG", "ERROR", "FATAL" ("INFO", "NOTSET", WARN, "WARNING")

3.4 TwoDGraph

class TwoDGraph.**Graph**(*rows*, *title='YT Display'*)

Create a window in which to display the YT display

The window contains two grids and the YT display. The axes are time, and amplitude.

Parameters **rows** (*int*) – Defines the horizontal size of plot in samples.

Keyword Arguments **title** (*str*) – The name of the window. Defaults to *YT Display*

np = <module 'numpy' from '/usr/lib/python3.8/site-packages/numpy/__init__.py'>

pg = <module 'pyqtgraph' from '/usr/lib/python3.8/site-packages/pyqtgraph/__init__.py'

update(*data*)

Obtain next chunk of data, update the waterfall display

3.5 ThreeDGraph

class ThreeDGraph.**Graph**(*rows*, *cols*, *title='3D Waterfall Display'*, *distance=50*)

Create a window in which to display the 3d display

The window contains two grids and the 3d display. The axes are time (history of former traces) , time (for a given trace) and amplitude of the points in the trace. The grids are on the (history)time-amplitude and (history) time-time axes. No grid is generated for the (trace) time-amplitude axes so as not to occlude the line-of-site of the display

Parameters

- **rows** (*int*) – Defines the size of the waterfall in rows. This value represents the numper of points per trace.

- **cols** (*int*) – Defines the size of the waterfall in columns. This value represents the depth of the storage. This is the number of traces to display

Keyword Arguments

- **title** (*str*) – The name of the waterfall window. Defaults to *3d Waterfall Display*
- **distance** (*int*) – The inital distance of the observer from the 3D display

np = <module 'numpy' from '/usr/lib/python3.8/site-packages/numpy/__init__.py'>

pg = <module 'pyqtgraph' from '/usr/lib/python3.8/site-packages/pyqtgraph/__init__.py'

update(*data*)

Obtain next chunk of data, update the waterfall display

CHAPTER

FOUR

LICENSE

This program is licensed under the MIT license.

CHAPTER

FIVE

THIRD PARTY SOFTWARE

The following modules, not provided by Python or the *Siglent Waterfall* repository, are required at run time and are installed automatically by *pip*:

Module Name	Version	Description
Python	3.8	Interpreter
Click	7.0	Command line interface
numpy	17.4	Vector Math
PyOpenGL	3.1.0	3D graphics engine
PyQt5	5.13.2	GUI subsystem
PyQt5-sip	12.7.0	GUI subsystem
pyqtgraph	0.10.0	graphing package
pyVISA	1.10.1	Instrument control
scipy	1.3.3	Vector math

This program does use *Sphinx* to automatically generate documentation from the *doc* strings in the Python code itself. Sphinx can create man files, HTML files, pdf files, and others. This program uses the HTML documentation for the web application and pdf files for written manuals. It also generates man files for reference when running on Linux from a command line.

The following top level modules are required to build the documentation for this program suite. These modules may have dependencies of their own and are system dependent. These are not automatically installed.

Module Name	Version	Organization	Description
Sphinx	1.8.1	http://www.sphinx-doc.org/	Documentation Generator
texlive	2018.48568	http://tug.org/	Web Template Engine

To build the HTML documentation, from the root directory, run:

make html

To build the pdf documentation, from the root directory, run:

make latexpdf

To build the *man* documentation, from the root directory, run:

make man

CHAPTER

SIX

INDICES AND TABLES

genindex modindex search

PYTHON MODULE INDEX

INDEX

Symbols

A

B

G

L

M

N

P

S

T

U