

# GazeAtDS1054 - a RIGOL DS1054Z viewer

Got tired gazing at my scope, trying to read the small numbers on the display. So this little program was meant to display the data nice and big on my PC screen. The more the project turned out to be fun, the more it got out of hand. But to be clear: Nothing is done here that the DS1054 could not do - but all is done a little bit bigger and a little bit different. In short: An other Rigol-Viewer - this time for Linux based OS's. Have a look at what it does and if you find it could be useful to you, be welcome to download and use it:

* free of charge (see license )
* free of guarantee (see the license)
* free of support (also see the license)

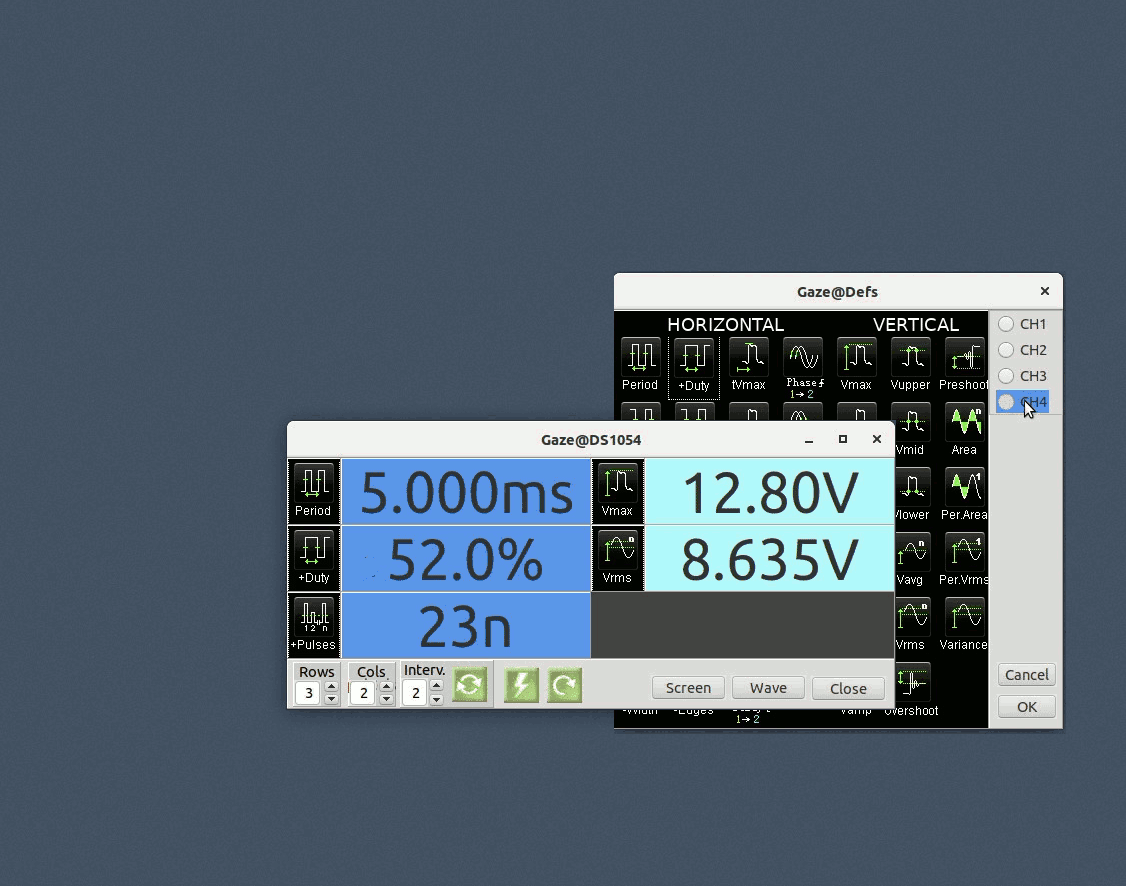
## What does it do

GazeAtDS1054: A viewer for the RIGOL DS1054Z (though according to the RIGOL specs, it should work with all MSO1000Z/DS1000Z series digital oscilloscopes) for Linux distros. It captures data from the oscilloscope via USB or TCP and:

* displays selected measurements in a free definable grid form
* displays and exports screen-shots from the scope display
* displays and exports wave-, protocol- and FFT-data in LowRes or HighRes mode. Protocol data can also be exported and viewed in Pulseview.

Data from the scope can be acquired in two ways:

* LowRes mode: This downloads the scope display which is 1200 pts wide.
* HighRes mode: This downloads the scope memory (or part of) which can be up to 12M pts per channel. HighRes downloading can take quite a while depending on the connection (USB or TCP) and the selected data range.



## What are the requirements

This application runs on Linux distros. It was developed on Ubuntu with FPC-Lazarus. It accesses the scope via Python/PyVisa using P4D components.

If protocol analyses will be done, Sigrok-cli and/or PulseView have to be installed

Various versions of GazeAtDS1054 were tested on the following 64bit Linux distros:

* ubuntu-18.04.5-desktop-amd64 (installed and live USB)
* ubuntu-20.04.2.0-desktop-amd64 (installed and live USB)
* debian-live-10.9.0-amd64-gnome+nonfree (live USB)
* Fedora 34 (Workstation Edition)
* Manjaro -gnome-21.0.3

and the Python 64bit versions:

* python 3.6
* python 3.8
* python 3.9

and the Sigrok 64-bit versions of:

* sigrok-cli 0.7.0
* sigrok-cli 0.7.2
* PulseView 0.4.0
* PulseView 0.4.2

You might be out of luck with Ubuntu versions < 18.04 or Python versions < 3.6. But chances are good, it runs on newer versions of other Ubuntu flavors or Debian derivatives.

## Install it

If you are running a Debian derivative like Ubuntu with Python3 the installation is straight forward. Make sure Universe Repository is enabled. Then:

$ sudo apt update  
$ sudo apt install <file-location>/gazeatds1054\_x.x.x-x\_amd64.deb  
$ # Install Sigrok if you are going to use protocol analyser  
$ sudo apt install sigrok   
$ # Please reboot the system now.  
$ # Or, if you don't want to reboot, reloading the rules should suffice:  
$ sudo udevadm control --reload-rules  
$ sudo udevadm trigger

Done! Read on for installation on other OS configurations.

In this Documentation it is assumed, that a single Python3 version is installed and used. Thus in all commands Python is referred to as "python3". If more than one python version is installed, make sure you are referring to a specific version by using the minor version in commands (e.g. "python3.9" or "python4.2" etc)

If a working version of PyVisa is already installed on your system, you want to Check the connection to the scope first (see below). If this is successful, it might be better to just unzip the application files instead of installing yet another version of PyVisa.

This application uses gtk2. If, on some of the leading edge distros, a higher version of gtk is used, gtk2 can be installed like:

$ sudo apt install gtk2

Suggestion: Boot up a Live version of your favored OS to test Gaze@DS1054. That's how I tested the application.

Find more install options in section More... below.

## Run it

Run GazeAtDS1054

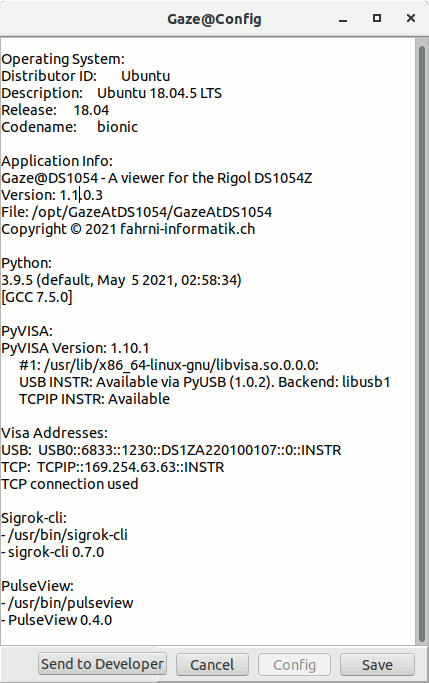
* from the Ubuntu launcher or
* from Run Command Dialog ALT+F2 type:

/opt/GazeAtDS1054/GazeAtDS1054

There is no extensive Help system coming with GazeAtDS1054 nor is there a Users Guide. Try it and you should get what you click and hope for. Hovering over a button might give you a Hint or typing F1 might display a short HELP for you.

#### Configuration

The first time GazeAtDS1054 is started up, a configuration dialog is displayed.



You will be asked to confirm/modify

* the Python library to be used. You can find your Python versions on the terminal:

$ python3 --version #or python4 --version   
Python 3.6.9   
$ find / -type f -iwholename "/usr/\*libpython3.6\*.so\*" 2>&1 | grep -v 'config\|Permission'   
/usr/lib/x86\_64-linux-gnu/libpython3.6m.so.1.0

* the USB/LAN Visa addresses for the scope If a USB connection to the scope is available, the addresses are automatically filled in Else you have to type in the addresses manually the LAN Visa Address being available from the IO.Settings of the scope and the USB Visa Address usually being 'USB0::6833::1230::DS1ZA220100107::0::INSTR'
* The connection Type USB or TCP you wish to use. This can be changed later using the Run option --change-input-type. Note: If TCP is used, the USB cable has to be removed.
* The path to the Sigrok-cli application if you intend to analyse protocol data.
* The path to the PulseView application if you intend to analyse protocol data with PulseView.

Finally, you can send your configuration details to the developer (yes, that's me!). It would be appreciated a lot, give me an additional kick to continue bothering about the project instead of growing lonely and having sour feelings.

#### Run options

You can open the configuration dialog using the Run Command Dialog ALT+F2 and type:

$ /opt/GazeAtDS1054/GazeAtDS1054 --reset

You may change (toggle) the connection type USB/LAN using the Run Command Dialog ALT+F2 and type:

$ /opt/GazeAtDS1054/GazeAtDS1054 --change-input-type

The installation log can be displayed. Start the Run Command Dialog ALT+F2 and type:

$ /opt/GazeAtDS1054/GazeAtDS1054 --info

## More ...

#### Manual Installation

If you want to install GazeAtDS1054 manually (your Linux distro can't handle deb-packages or Python4 is being used etc), all application files are available in gazeatds1054\_x.x.x-x\_amd64.zip and the following terminal commands might get you there:

$ sudo apt update  
$ sudo apt install python3-pip  
$ python3 -m pip install pyvisa  
$ python3 -m pip install pyvisa-py  
$ python3 -m pip install pyusb  
$ sudo unzip <PathToZip>gazeatds1054\_x.x.x-x\_amd64.zip -d /  
$ sudo groupadd usbusers  
$ sudo usermod -aG usbusers $USER  
$ sudo usermod -aG dialout $USER  
$ mkdir /home/$USER/.config/GazeAtDS1054  
$ chmod 755 /home/$USER/.config/GazeAtDS1054  
$ mkdir /home/$USER/.config/GazeAtDS1054/sigrok  
$ chmod 755 /home/$USER/.config/GazeAtDS1054/sigrok  
$ echo 2 > /home/$USER/.config/GazeAtDS1054/sigrok/version  
$ # If you don't want to reboot now, do this:  
$ sudo udevadm control --reload-rules  
$ sudo udevadm trigger

#### Additional Users

If GazeAtDS1054 should be used by other users on the system, the local environment for every user must be set up as follows:

$ python3 -m pip install pyvisa  
$ python3 -m pip install pyvisa-py  
$ python3 -m pip install pyusb  
$ sudo usermod -aG usbusers $USER  
$ sudo usermod -aG dialout $USER  
$ mkdir /home/$USER/.config/GazeAtDS1054  
$ chmod 755 /home/$USER/.config/GazeAtDS1054

#### Remove Installation

$ sudo deluser $USER usbusers  
$ # delete all files listed by zipinfo  
$ zipinfo -1 <DirToZip>/gazeatds1054\_x.x.x-x\_amd64.zip   
$ rm -R /home/$USER/.config/GazeAtDS1054  
$ # if $USER does not need to be member of usbusers, remove it  
$ sudo deluser $USER usbusers  
$ # if pyvisa is not used anywhere else, remove it  
$ python3 -m pip uninstall pyusb  
$ python3 -m pip uninstall pyvisa-py  
$ python3 -m pip uninstall pyvisa  
$ sudo apt remove python3-pip

## It does not work!

Be warned: As you probably know, finding out why an installation doesn't work can be a tedious undertaking, especially with different software versions installed. So, if you are not a nerd or a pensioner with unlimited time at hand, it might be wise to consider other possibilities like squinting your eyes again while gazing at your scope.

However, before giving up, you might want to try the following:

#### Check the application installation log

If you actually got GazeAt1054 up and running, the installation log can be displayed. Start the Run Command Dialog ALT+F2 and type:

$ /opt/GazeAtDS1054/GazeAtDS1054 --info

#### Check your installation

You can do some basic checks on your pyvisa installation:

* you should have a Python version >3.2 installed.
* make sure you got the 64bit Ubuntu and Python versions software
* the backends for USB INSTR and TCPIP INSTR must be available

$ python3 -m visa info  
Machine Details:  
 Platform ID: Linux-5.8.0-50-generic-x86\_64-with-glibc2.29  
 \*\*Processor: x86\_64\*\*  
  
Python:  
 Implementation: CPython  
 Executable: /usr/bin/python3  
 \*\*Version: 3.8.5\*\*  
 Compiler: GCC 9.3.0  
 \*\*Bits: 64bit\*\*  
 Build: Jan 27 2021 15:41:15 (#default)  
 Unicode: UCS4  
  
PyVISA Version: 1.11.3  
  
Backends:  
 ivi:  
 Version: 1.11.3 (bundled with PyVISA)  
 Binary library: Not found  
 py:  
 Version: 0.5.2  
 ASRL INSTR:  
 Please install PySerial (>=3.0) to use this resource type.  
 No module named 'serial'  
 \*\*USB INSTR: Available via PyUSB (1.1.1). Backend: libusb1\*\*  
 USB RAW: Available via PyUSB (1.1.1). Backend: libusb1  
 \*\*TCPIP INSTR: Available\*\*   
 TCPIP SOCKET: Available   
 GPIB INSTR:  
 Please install ...

#### Check the connection to the scope

Running Python in a Terminal, you should be able to connect to your scope. Note: Disconnect USB cable it you are using the Ethernet connection.

$ python3  
>>> import pyvisa as visa  
>>> rm=visa.ResourceManager("@py")  
>>> print(rm.list\_resources())  
('USB0::6833::1230::DS1ZA220100107::0::INSTR',)  
>>> scope=rm.open\_resource('USB0::6833::1230::DS1ZA220100107::0::INSTR')  
>>> scope.query("\*IDN?")  
RIGOL TECHNOLOGIES,DS1054Z,DS1ZA220100107,00.04.04.SP4\n  
>>>

#### A word about Sigrok

The Sigrok decoder library spots different option parameter-names between lib-versions. I setteled for libsigrokdecode\_0.5.3. If analysing protocols with Gaze@Wave does not work, you can try to use PulseView or check on your decode libray as follows:

$ cat /usr/share/libsigrokdecode/decoders/uart/pd.py | grep num\_data\_bits

If this finds any occurences of num\_data\_bits, then you got an older library version. You can try to install a newer version of the library or of Sigrok altogether. I circumnavigated the problem by patching the Sigrok installation, downloading libsigrokdecode\_0.5.3.orig.tar.gz and extract the decoders folder to ~/.local/share/libsigrokdecode/ . Check the lib like so:

$ cat ~/.local/share/libsigrokdecode/decoders/uart/pd.py | grep "'data\_bits'"

If this gives you a valid output, you should be good to go. Good luck!