GeigerLog Manual

by ullix

Version 0.9.92

August 2019

What's New in GeigerLog 0.9.92 ?

- The Gamma-Scout counter are now also supported in GeigerLog

What's New in GeigerLog 0.9.91 ?

- after some bugfixes PyQt5 is used as sole toolkit for all systems, further reducing installation headaches
- some more convenience functions (creating CSVs, printing)

What's New in GeigerLog 0.9.90 ?

- All data are now stored as database files
  - more compact
  - much faster
- A Windows installation has become much, much simpler thanks to the latest graphic toolkit PyQt5 fully supported in GeigerLog
- More supported devices – 3 times more than before:

- GMC-Devices	– as before
- AudioCounter Devices	– counting the audio clicks of Geiger counters
- I2C-Devices	– environmental variables
- RadMon Devices	– as before
- AmbioMon Devices	– Geiger counts plus environmental variables
- LabJack Devices	– environmental variables

- More variables available – now 12 variables for recording
- Reading any CSV file to transform into a GeigerLog database
- Flexible, transient graph scaling with powerful interpreter handling algebraic formulas with mathematical functions (log, trig, sqrt, power,…) to correct offsets, linearize values, convert air-pressure to sea level, account for dead time losses of Geiger counts, and more
- More Quality Control means with a Scatter Plot for data, and printing the values of variables selected in the graph
- Overall improved handling of graph, statistics, analysis

Recommended Reading on the subject from the same author:

All available on the SourceForge site: https://sourceforge.net/projects/geigerlog/

GeigerLog - Potty Training for Your Geiger Counter

This article is about the use of natural Potassium to give your Geiger counter a little bit of a training workout when you get tired of measuring just the background. Potassium is omnipresent on the earth, essential for all life, may already be available in or around your home or garden, and has a little bit of natural radioactivity – though well below any danger zones.

I will show how to best use it, taking advantage of today's Geiger counter technology and software.

GeigerLog - Going Banana

Ever heard the term 'banana equivalent dose'? It refers to the Potassium content of bananas, which gives the bananas a tiny little bit of radioactivity. Nevertheless, I demonstrate that you can measure this with a Geiger counter, but it is tricky as the activity is very low and demands in-depth statistical considerations.

GeigerLog - Review Smart Geiger Pro (SGP-001)

The **Smart Geiger Pro (SGP-001)** is a semiconductor detector for radioactivity, i.e. it is NOT using a Geiger-Müller tube! While it is designed to plug into the headphone plug of a smartphone, the present GeigerLog version 0.9.90 allows to use it connected to a personal computer.

GeigerLog - AudioCounter-Support

Some Geiger counters – especially very old ones and modern low-cost varieties – generate audio-clicks for each registered radioactive event. But even the very modern semiconductor based radioactivity detector **Smart Geiger Pro (SGP-001).** GeigerLog now fully supports those audio counters. In the article a GMC-300E+ counter, connected digitally and via audio simultaneously, demonstrates that the results are valid.

Author	ullix
Credits	Phil Gillaspy for extended documentation of Geiger counter commands
	GQ Electronics LLC for documentation
Copyright	
License	

Table of Contents

Overview

GeigerLog is a combination of data **logger**, data **presenter**, and data **analyzer**.

It is based on **Python (Version 3)**, hence it runs on Linux, Windows, Macs, and other systems.

GeigerLog had initially been developed for the sole use with Geiger counters, but has now become a more universal tool, which equally well handles environmental data like temperature, air-pressure, humidity, and light, and is ready for future sensors. In its present state it can e.g. be deployed as a monitor for a remote weather station, complemented with a Geiger counter to monitor radioactivity.

The most recent version of GeigerLog, including this manual, can be found at project GeigerLog at SourceForge: https://SourceForge.net/projects/geigerlog/.

Currently Supported Devices

GMC Devices:

GeigerLog continuous to support GQ Electronics's [1]) **GMC-3xx**, **GMC-5xx**, and **GMC-6xx** line of classical Geiger counters, including the variants with an additional 2nd Geiger tube.

Specific to these devices is that they can store up to several weeks of recordings in their internal memory. GeigerLog can read this internal memory.

***New*: AudioCounter Devices:**

Any Geiger counter which produces audible clicks that can be fed into a computer via microphone-in or line-in can now be recorded and logged by GeigerLog. Many low-cost Geiger counters produce only audio-clicks. They can now be used with GeigerLog. This also provides an alternative way to connect the **GMC** counters, e.g. in case their USB connection fails

In particular, GeigerLog now allows to use the interesting **Smart Geiger Pro (SGP-001)** (pdf) semiconductor Geiger counter with a Personal Computer, which has so far not been possible!

***New*: I2C Devices:**

GeigerLog can now handle I2C based sensors. Presently implemented is the use of the ELV dongle connected with a BOSCH BME280 sensor (temperature, air-pressure (atmospheric-pressure), and humidity) and the TSL2591 light sensor. More information at I2Cpytools at Sourceforge (https://sourceforge.net/projects/i2cpytools/).

RadMon Devices:

Support of the **RadMon+** [2]) hardware, which can provide a Geiger counter as well as an environmental sensor for temperature, air-pressure (atmospheric-pressure), and humidity.

These devices acts as IoT (Internet of Things) devices, and transmit their data wirelessly

***New*: AmbioMon Devices:**

Support of the **AmbioMon** [3]) hardware, which can provide a Geiger counter as well as an environmental sensor for temperature, air-pressure (atmospheric-pressure), and humidity.

These devices acts as IoT (Internet of Things) devices, and transmit their data wirelessly

1 GQ Electronics LLC**,** 5608 Delridge Way SW, Seattle, WA 98106, USA, http://www.gqelectronicsllc.com/
2 DIYGeigerCounter https://sites.google.com/site/diygeigercounter/
3 This device is in development and not yet publicly available

***New*: LabJack Devices:**

Support of the **Labjack** (https://labjack.com/) hardware U3 in combination with the ei1050 probe for temperature and humidity.

***New*: Gamma Scout Devices:**

Full support of the **Gama-Scout** (https://www.gamma-scout.com/en/) devices ***Standard***, ***Alert***, ***Rechargeable,*** and partial support of model ***Online*** (see Gamma-Scout Devices on page 42) .

Main Operations – Logging, Displaying, Analyzing

Logging will be done with a user defined cycle time of 0.1sec or longer. Each logging cycle consists of

1. reading from the connected devices
2. saving the data into a database file
3. printing the data as a numeric values to the screen
4. and displaying the data as a live graph, auto-updating after each log cycle

Comments can be added to the log file before, during, and after logging.

Displaying means that the data are shown as a Time Course graph, i.e. as a plot of value versus time. The graph uses two Y-axis:

The **left Y-axis** is reserved for Geiger counter data, and is shown in units of CPM / CPS or µSv/h. The **right Y-axis** is reserved for environmental data. If temperature data are shown, the choice of units is between °C and °F.

To display variables with very different numerical values on a common scale – like temperature (e.g. 0 … 30°C) and air-pressure (e.g. 970 … 1030 hPa) – the variable values can be scaled for plotting, e.g. here by subtracting 1000 from the air-pressure. The saved value will NOT be affected.

All scales are set automatically, but can be changed manually.

Time ranges can be set to plot data only within that range and to limit any quality control analysis to only those data. These ranges can be entered manually or by left/right mouse clicks. The time can be shown as Time-of-Day, or time since first record in units of sec, min, hours, days, or auto-selected in auto mode. The graphs can be stretched, shifted, and zoomed for details, and saved as pictures in various formats (png, jpg, tif, svg, ...).

Analyzing is supported with several **Quality Control** tests, which can be applied to the data. Beyond the standard statistical properties – as a brief summary or a more elaborate statistics – a Poisson test can be applied to see if the Geiger counter data are valid at all, and how well they fit to a Poisson distribution. Also, a FFT frequency and Autocorrelation analysis by Fast Fourier Transform (FFT) can be done to check for any cyclic effects in any of the measured variables.

All manipulations of the plots, and all data analysis can be done during ongoing logging without disturbing it.

Supporting Data Files

Several genuine and synthetic recordings of Geiger counter and environmental data are included, among them a recording from an international long-distance flight. The synthetic data can help greatly to understand the data produced by a Geiger counter.

Introduction to GeigerLog

Installing and Starting GeigerLog

GeigerLog requires a **Python 3** environment. It will NOT run on Python2 [4]) !

It was developed with Python version 3.5.2, and verified to run with Python 3.4, 3.6, and 3.7. In addition to a Python3 environment a few Python modules are needed, which generally are not available in a default installation.

GeigerLog uses the modern PyQt5 toolkit; the older PyQt4 toolkit is longer supported.

Step-by-Step installation instructions for Python on **Linux, Windows and Mac** are provided in Appendix H – Installation beginning on page 75.

The Software

The software comes in a zipped package containing the Python scripts and associated resources like icons and manual.

The package is named **`geigerlog-scripts-vXYZ.zip`** (xyz is the version number).

Installing

Download the package and unzip into a directory of your choice. It creates a directory 'geigerlog' (which will be your working directory), and subdirectories 'data' and 'gres' (GeigerLog resources).

Starting

1. Start GeigerLog with:
   `/path/to/geigerlog`
2. If Python is not in your path, you may have to start GeigerLog with:
   `python /path/to/geigerlog`

If it does not work, note any error message and look into Appendix H – Installation on page 75.

Look & Feel

The Python software depends on the host computer for the Look & Feel. If GeigerLog does not look the way you like it, see Appendix A – Look & Feel on page 58.

Default Configuration

GeigerLog's default configuration is to use the GMC counters and the AudioCounters. This can be changed in the configuration file `geigerlog.cfg` (more on this in chapter Configuration of GeigerLog on page 20).

4 The last GeigerLog version running on Python version 2.X is 0.9.06. Use this if you can't use Python3, but upgrading to Python3 is strongly suggested!

The GeigerLog Window

GeigerLog has a single window with predefined usage areas. Fig. 1 gives an overview of the GUI.

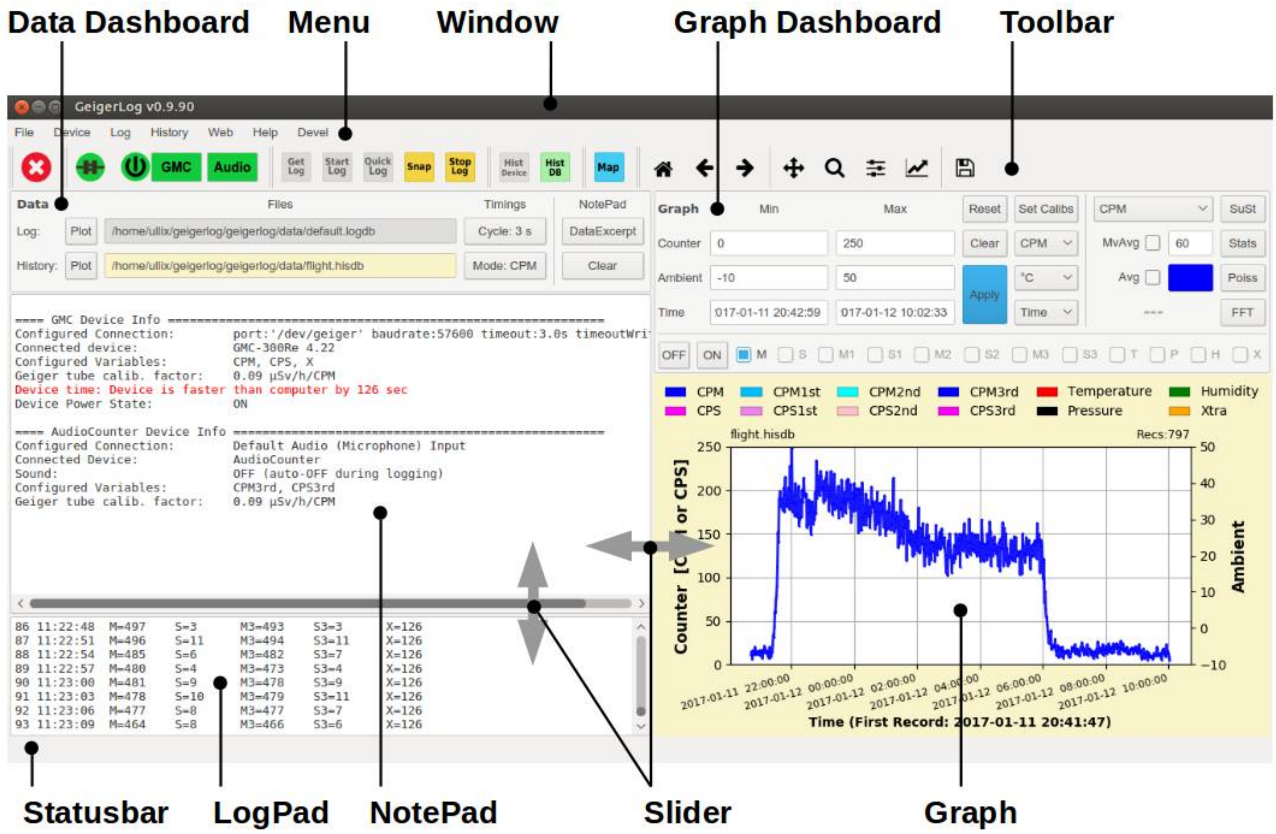


Fig. 1 GeigerLog Window with Annotations

Window: The start-up window is sized to fit on a standard screen of 1366x768 pixel. GeigerLog will run on a 1024x768 screen (see configuration file `geigerlog.cfg`), but it is cumbersome to use.

Menu: Menu items may be grayed out when currently not selectable. Some items have keyboard shortcuts in the form of CTRL-X; see the menus for the codes to be used in lieu of the X.

Toolbar: A toolbar with icons for quick mouse access to the more frequent actions.

Statusbar: The bottom line of the window holds the Statusbar providing info about an item when you point the mouse cursor over the item. Some error messages are also shown here.

Data Dashboard: Manage Log and/or History files, Timings, and NotePad.

NotePad: A scratch-pad type of area for various textual and numeric information.

LogPad: During logging you find here all log values since last start of logging.

Graph Dashboard: Settings to configure your graph.

Graph: Graphs will be shown here.

Slider: change the size of the sub-windows with your mouse to make space where you need it

Quick Tour of GeigerLog

GeigerLog is best shown with a quick demo guiding you through typical usage steps. GeigerLog can be run without a connected device in order to analyze **existing** data:

With GeigerLog running, click on menu 'History' → 'Get History from Database' and select **flight.hisdb**. The original data from an international flight from Germany to the Maldives will be loaded and displayed as a graph, showing the Time Course of CPM versus Time-of-Day.

In the Graph Dashboard, click the drop-down button currently showing 'CPM', and select 'µSv/h'. The graph changes, now showing µSv/h versus Time-of-Day. Now select the Time Unit drop-down button currently showing 'Time' and select 'auto'. The graph switches to µSv/h versus time-since-first-record in the automatically selected unit 'hours'.

In the graph, do a mouse-**left**-click somewhere on the vertical line near 8 h, and a mouse-**right**-click on the vertical line near 10 h. Note that the Time Min and Max fields in the Graph Dashboard are filled by the mouse clicks. Click the Apply button. The graph is zoomed-in to the descending part of the flight from about time 8 h to 10 h. You can fine tune the range with further mouse clicks, or manually edit the Time Min and Max fields, clicking Apply after changes.

Click the check button under 'MovAvg(s)'. A Moving Average is shown as an overlaid yellow-frramed-line, with an averaging period of 60 sec. Since the data were collected by the Geiger counter in the 'CPM, Saving every minute' mode, which is already the average over 60 sec, no effect will be seen. Change the 60 to 600. The graph will update automatically. Now the data are averaged over 10 minutes, equal to 10 data points. Try entering other numbers than 600.

Click the 'Clear' button in the Data Dashboard on the left side, then click the 'DataExcerpt' button. Data from the beginning and the end of the flight will be printed into the NotePad.

Click the 'SuSt' (Summary Statistics) button in the Graph Dashboard. Some brief statistics is printed into the NotePad. Click the 'Stats' button for a more detailed statistics in a pop-up windows. Click 'Reset', then 'hour' under Time Unit, then mouse-**left**-click on the vertical line near 10 h, and Apply. Then click button 'Poiss', and a 'Histogram with Poisson Fit' will be shown in a new window together with some further statistics. Click 'OK' to close. Click button 'FFT' to see an FFT analysis of the count rate data (explained later). Click 'OK' to close.

On the toolbar click the right-most icon to save the current graph as an image file. The availability of image formats depends on your computer, but typically png, jpg, tif, and svg is available.

Click the Reset button to reset the graph to starting conditions.

The data nicely show that the background radiation, of which a good part is cosmic radiation, increases when going from ground level up to airplane cruising altitude, and up there decreases going from northern latitudes towards the equator. This is known since early last century. But at that time the radiation measuring devices had a weight of a ton mounted on a ship; today you can carry them in your shirt pocket while traveling by airplane! (Yes, the counter can be taken into the cabin.)

Running GeigerLog

This chapter explains the general approach; considerations for a specific device will be dealt with in a later chapter.

Establishing a GeigerLog Connection to Your Devices

To have GeigerLog interact with your device, you must **establish a connection** between them. This has two requirements: The first is the hardware between the device and the computer, the second is the software activation within GeigerLog.

The **hardware** could be based on a wire, like a USB cable or an Audio cable, or it could be based on a wireless connection, like WLAN (also called WiFi). The **software** requirement is that the GeigerLog configuration file `geigerlog.cfg` is properly defined for the devices you will use, and that you have selected the menu command: **Device** → **Connect Devices**. This last action establishes the needed connection to the activated devices. Instead of using the menu command you could use the more convenient **Toggle Connection** button in the toolbar, the left-most icon with a plug symbol in the next figure.

Fig. 2 The Device Toolbar signaling the Connection Status when all possible devices are activated in the configuration file. Top: Before -, Bottom: After – establishing a connection

The green devices are successfully connected, the red ones failed (here the IoT server was down)

The device icons turn green upon a successful connection to that device, red otherwise. Red devices are NOT available for logging; you may need to verify your configuration file `geigerlog.cfg`.

In addition to the icon color changes, some info will be printed for each of the devices into the NotePad similar to this one (printout here limited to one successful and one failed connection) :

```
==== GMC Device Info ===========================================================
Configured Connection:      port:'/dev/geiger' baudrate:57600 timeout:3.0s
Connected device:           GMC-300Re 4.22
Configured Variables:       CPM, CPS, X
Geiger tube calib. factor: 0.0065 µSv/h/CPM
Device time: Device is faster than computer by 11 sec
Device Power State:         ON

==== Connect RadMon Device =====================================================
ERROR: Connection failed using server IP='iot.eclipse.org', port=1883
ERROR: Message: '[Errno 111] Verbindungsaufbau abgelehnt'
RadMon+ not connected. Verify server IP and server port
Failure to connect with Device: RadMon+
```

You can repeat these info printouts any time by clicking on the device-named icons.

You are now ready to start logging!

Device Mappings

Before you start logging, take a look at the Device Mappings**.** With the many device types now supported by GeigerLog, and the many variables available for recording, it is important to make sure that no variable is written to by more than one device! The device mapping is shown in the NotePad upon connecting, and can be called from the menu **Device** → **Show Device Mappings**.

The next paragraph shows Device Mappings after a successful connection with no mapping problem:

```
==== Device Mappings ==========================================================
The configuration is determined in the configuration file geigerlog.cfg.

Device    :  CPM CPS CPM1st CPS1st CPM2nd CPS2nd CPM3rd CPS3rd T   P   H   X
---------------------------------------------------------------------------
GMC       :  X   X   –      –      –      –      –      –      –   –   –   –
Audio     :  –   –   –      –      –      –      X      X      –   –   –   –
RadMon    :  –   –   –      –      X      –      –      –      X   X   X   –
Mapping is valid; no problems found
```

The GMC-Device counter collects Geiger counts at CPM and CPS, the AudioCounter at CPM3rd and CPS3rd, and the RadMon Geiger counts at CPM2nd, and the RadMon temperature, pressure, and humidity at T, P, and H. There are no conflicts.

However, the next example shows multiple duplicate mappings, highlighted in red:

```
==== Device Mappings ==========================================================
The configuration is determined in the configuration file geigerlog.cfg.
ALERT: Mapping problem of Variables
Variable CPM3rd is mapped to more than one device
Variable T is mapped to more than one device
Variable P is mapped to more than one device
Variable X is mapped to more than one device

Device    :  CPM CPS CPM1st CPS1st CPM2nd CPS2nd CPM3rd CPS3rd T   P   H   X
---------------------------------------------------------------------------
GMC       :  X   X   –      –      –      –      –      –      –   –   –   X
Audio     :  –   –   –      –      –      –      X      X      –   –   –   –
RadMon    :  –   –   –      –      X      –      –      –      X   X   X   –
AmbioMon  :  –   –   –      X      –      –      X      –      X   X   –   X
Measurements are made on devices from top to bottom, and for each from left to
right. If double-mapping of variables occurs, then the last measured variable
will overwrite the previous one, almost always resulting in useless data.
```

As any variable, which is measured later in the log cycle, overwrites any previously measured one, there will generally be nonsense generated with such a mapping. Correct mapping in the GeigerLog configuration file `geigerlog.cfg`.

Logging with GeigerLog

Once a connection is established, you can start logging.

> NOTE: The GMC Geiger counters are so far the only devices, which – beyond the logging mode – support another operating mode: **History**. This means reading their internal memory. It will be explained in a later chapter specific to the GMC devices.

Logging means that GeigerLog gets fresh data from the devices, saves them in a database file, prints them on the screen, and plots them to a configurable graphic. GeigerLog then waits for the user specified cycle time before it repeats the process.

This cycle time is set by clicking the **Timings Cycle** button in the **Data Dashboard**. A pop-up box allows you to enter a new cycle time of at least 0.1 seconds. A shorter cycle time cannot be entered. (see Manage Your Recordings – Data Dashboard on page 14).

While logging is ongoing, the cycle time cannot be changed!

Fig. 3 The Logging Toolbar's various stages

Top:	*Not logging, no log file loaded*
Middle:	*Not logging, a log file is loaded*
Bottom:	*Logging is ongoing*

Before you can log, a log file must be loaded, so click the **Get Log** icon in the toolbar to load an existing file, or define a new one. The toolbar will change and now also offer the **Start Log** icon. Click it to start logging. The toolbar will change again and allow only to stop the logging (Snap will be explained shortly). Other functions, which would interrupt logging, like exiting GeigerLog or loading the History from a GMC counter, are also disabled during logging.

The **Quick Log** icon saves you a step by automatically using the log file `default.logdb`. However, note that this file is overwritten every time you click Quick Log! If you want to attach data to a previous Quick Log recording, click Start Log instead. Quick Log is convenient if you want to just see current values, and don't care much about keeping the data.

Sometimes you may want to see fresh data right away and not wait for the next cycle. Simply click the **Snap** icon, and GeigerLog snaps a fresh record out of order and prints it into the NotePad. Snapped records are also saved in the log database just like any other record.

The result may look like this (it uses shortcuts as in the Graph Dashboard: M for CPM, S for CPS, T for Temperature, P for Pressure, H for Humidity, and X for Xtra):

```
==== Snapped Log Values =========================================
47 19:22:38  M=149 S=1 T=21.7  P=1000.29  H=41.0  X=76.2
```

A note on the logging cycle when measuring Geiger counter data

The Geiger counter needs less than 1 ms (millisecond) [5]) to register and process an event which results in a count. When the counting is set to CPS (Counts per Second) the counter's firmware sums up all events during the last second and reports this as CPS. At background radiation level there is approximately only 1 count every 3...4 seconds on average. But even if the count rate were much higher than background, it obviously does not make sense to sample more often than 1 second to get the 'counts-per-**second**'. Likewise, when CPM is selected, the counts during the last minute are summed up. Hence you get all counts reaching the Geiger tube when the values are logged only once every minute.

However, this gets boring when you sit at the computer and wait for Geiger counter clicks; therefore I use a 3 second cycle time even for CPM logging just to "see some action" ;-). But for long time logging you might want to set this to 60 sec or longer, and perhaps use the Moving Average (see Visualize Your Recordings – Graph Dashboard) for further smoothing the data.

This **oversampling** – sampling more often than really needed – has consequences for certain properties of the data, see Quality Control - FFT – FFT & Autocorrelation Analysis on page 26.

But keep in mind that neither oversampling nor undersampling – e.g. measuring a CPM value only once every 10 min – has an impact on the validity of your measured averages as long as your setup and radioactive source does not change over time. They will all be the same! This follows from the properties of Poisson distributions.

Remember: if you have set a long cycle time, and are waiting impatiently for the next reading to come up, you can always press the **Snap** button and get a reading right away!

Manage Your Recordings – Data Dashboard

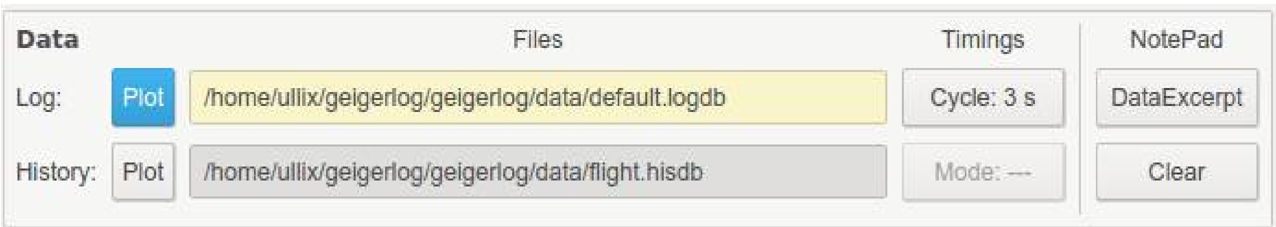


Fig. 4 Data Dashboard

The Data Dashboard lets you switch between viewing the Log file and the History file, and lets you set the timing for both. Also, you can print the data in numerical form to the NotePad.

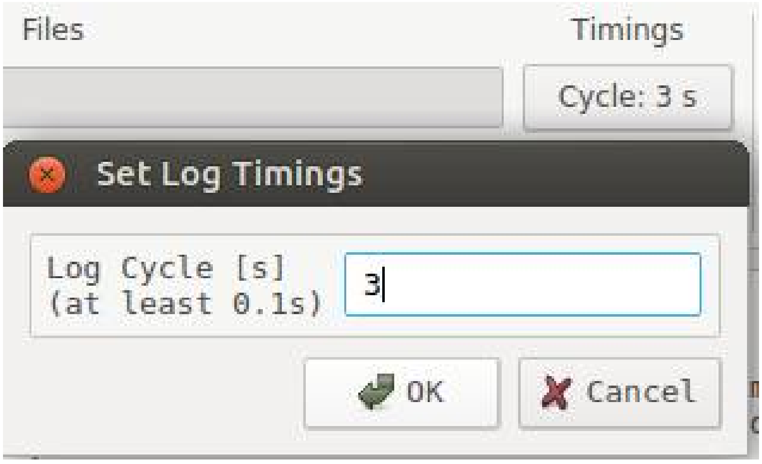
5 Based on measurements with an oscilloscope I determined the pulse length of an GMC-300E with M4011 tube or with SBM20 tube to be about 200µs, and with SBT11A tube about 150µs, as discussed in this post: http://www.gq-electronicsllc.com/forum/topic.asp?TOPIC_ID=4598 At 200µs the maximum count rate would be under CPS=5000. However, other effects, like microprocessor cpu power, , lower this even further.

Files

One **Log file** plus one **History file** can be loaded simultaneously. Their database filenames are shown, and they can be plotted – one at a time – using the **Plot** buttons. The file with the light yellow background is the one currently shown in the graph.

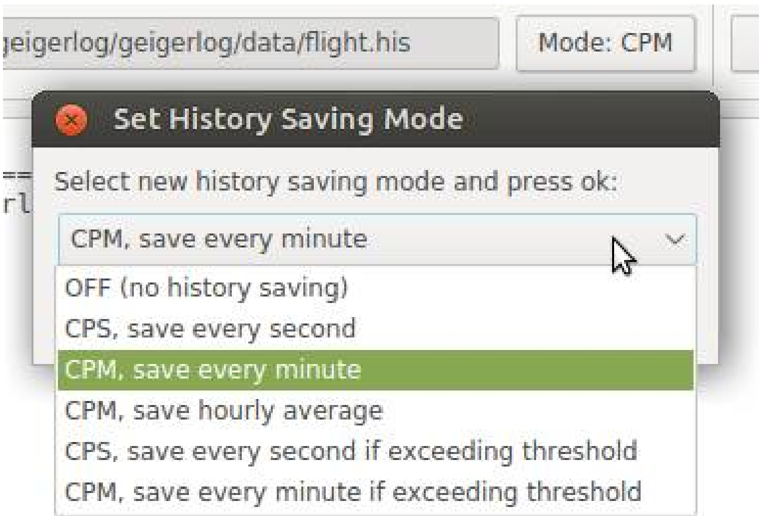
Timings

The buttons under **Timings** allow to set the logging cycle time and the History Saving mode.



Clicking the **Cycle** button opens a dialog box allowing you to enter a cycle time in seconds. Allowed is any number of at least 0.1 seconds; numbers less than 0.1 cannot be applied.

The cycle time can only be modified when logging is not active.



Clicking the **Mode** button opens a dialog box allowing you to chose between the available History Saving Modes.

For an explanation of the options see History Saving Mode on page 34.

NotePad

The buttons under **NotePad** provide convenience functions often used with GeigerLog.

The **Clear** button clears all content from the NotePad. The **DataExcerpt** button prints the currently active file – Log file or History file. If the file is short, the whole file is printed, if long, then printing is limited to only some first and last lines. Printing the full file irrespective of length can still be done via the menu, use: **Log → Show Log Data**, or **History → Show History Data**.

This button has a second function: when you print a full file from the menu, it may run for a long time when files are really large. The **DataExcerpt** button changes into a **STOP** button, allowing you to stop any such printout.

Visualize Your Recordings – Graph Dashboard

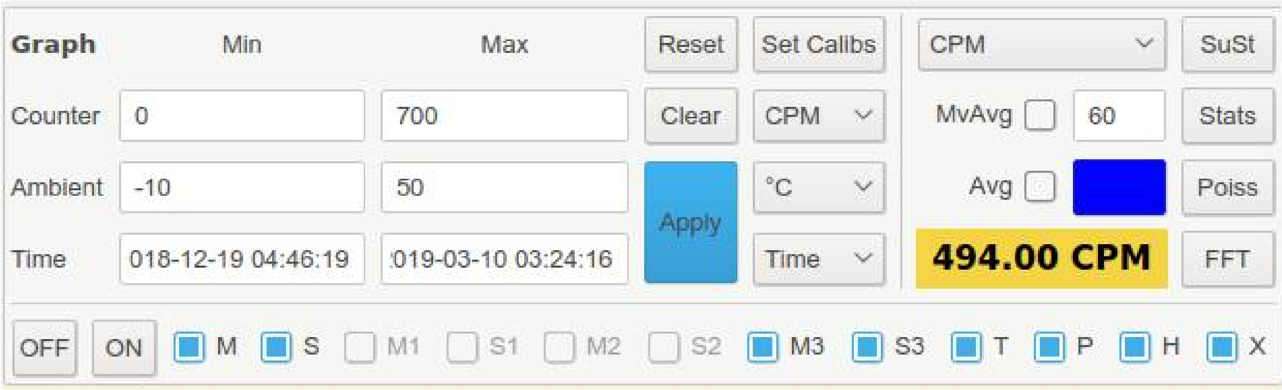


Fig. 5 Graph Dashboard

The Graph Dashboard controls what is displayed on the graph and how it it displayed. And no matter what you do here, the logging, downloading, processing or saving of the data will never be impacted!

The graph is laid out as Time Course of your data, i.e. the values of the variables are plotted versus time on the horizontal X-axis. It has two vertical Y-axis:

- the **left** Y-axis is labeled **Counter** and is used for all Geiger counter data
- the **right** Y-axis is labeled **Ambient** and is used for all environmental data, like temperature, air-pressure, humidity, light, and other.

What is displayed?

With up to 12 variables now available for display, it will often be important to reduce the number of variables displayed. The bottom row has buttons and checkboxes, which allow to show or hide a variable. Depending on active log or history, not all variables may be available. The checkboxes of unavailable variables are grayed out and cannot be selected.

The buttons OFF, ON switch all variables OFF, or ON, resp., (unavailable variables remain OFF and unselectable). The checkboxes use shortened names for the variables to ease the overview:

M	= CPM	from any Geiger counter device
S	= CPS	from any Geiger counter device
M1	= CPM1st tube	from any Geiger counter device
S1	= CPS1st tube	from any Geiger counter device
M2	= CPM2nd tube	from any Geiger counter device
S2	= CPS2nd tube	from any Geiger counter device
M3	= CPM3rd tube	from any Geiger counter device
S3	= CPS3rd tube	from any Geiger counter device
T	= Temperature	from any device yielding ambient data
P	= Air pressure	from any device yielding ambient data
H	= Humidity	from any device yielding ambient data
X	= Xtra	from any device yielding ambient data (e.g light, CO_2, ...)

Min/Max, Apply, Clear, Reset

The graph is auto-scaled in all 3 axis so that all data fit into the graph. However, the Min and/or Max value of the X-axis and both Y-axis can be set manually.

The Min/Max values for Counter and Ambient need to be entered from the keyboard. Those for the Time can be entered as e.g. ‘2018-07-18 14:00:41’. However, it is easier to use a mouse: with the mouse pointer resting in the graph, do a mouse-left-click to enter the Min Time value, and a mouse-right-click to enter the Max Time value.

To apply your entries to the graph, either click the **Apply** button or hit the **Enter** key.

To clear all entries in all Min/Max boxes, click the **Clear** button.

To reset all settings in the complete Graph Dashboard to their defaults, click the **Reset** button.

Set Calib[ration]s

Calibration refers to the conversion of the dose rate in CPM to the dose rate in µSv/h according to the formula:

Dose Rate [µSv/h] = <conversion factor [µSv/h/CPM] > * Dose Rate [CPM]

The **Set Calibs** button provides a convenience function for experimentation with different tubes. It allows to redefine the calibration (i.e. the conversion factor) currently in use for the up to three tubes of the Geiger counters. This redefinition can be applied at any time, but it is temporary and is discarded when the program ends, or a device re-activation occurs. For lasting changes please edit the GeigerLog configuration file `geigerlog.cfg`.

The same command is available from the menu **Device → Set Calibrations for Geiger Tubes**.

Units

X-axis: The time axis can display Time-of-Day or time-since-first-record. For the latter, set the unit selector to auto for an automatic choice between day, hour, minute, second, or set the time unit manually.

Left-Y-axis: This counter axis can either show CPM/CPS or µSv/h. If both a CPM and a CPS variable are shown at the same time, their 60fold difference may make the graph less informative. Deselecting one may be preferred.

However, when µSv/h is used, the two curves should overlap! The µSv/h data are calculated from the calibration factors as built into the firmware of the Geiger counters [6]). E.g., for the first tube (a M4011) of the GMC-300 series as well as the GMC-500 series this is 0.0065 µSv/h/CPM. For the second tube SI3BG in the 500+ this number is 0.194µSv/h/CPM. However, that latter number is clearly wrong [7]).

6 It is unknown for which condition exactly this applies; the calibration does depend on the type of radiation being measured, like gamma and beta, and their energies. Probably this conversion is valid **only** for gamma radiation with energies around 1 MeV; an attempt to justify this explanation is given in Appendix G – Calibration on page 73.

7 e.g. http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=5322

This setting can be changed in **Device** section of the configuration file geigerlog.cfg. In GeigerLog's default configuration the calibration factor for the 2nd tube is set to 0.48µSv/h, based on the ratio to the M4011 in an experiment of my own [8]).

Right-Y-axis: This Ambient axis is used for all environmental data, like temperature, air-pressure, and humidity. The temperature can be displayed in units of either °C or °F.

Note: Generally all data are displayed as recorded, but they can be scaled for display, while still being saved unmodified. One example is air-pressure, which is convenient to display as 'air-pressure minus 1000'. Since air-pressure is typically within the range of 970 … 1030 hPa, this transformation allows it to be displayed at the same scale as the other environmental variables. The values saved to the log file are always the unmodified original values.

Selected Variable

One of the displayed variables can be set to become the **Selected Variable** by selecting it in the drop-down box in the upper right corner of the Graph Dashboard. Only variables being displayed can be selected!

This **Selected Variable** will be highlighted in the graph with a brighter color and a thicker line, while the other variables will be dimmed. During logging, its last value will be shown in the **Last Value Box** as black letters on a golden colored background (in the figure it shows '494.00 CPM'). The other functions for analysis and quality control also work only on the **Selected Variable**.

If the checkbox **Avg** is checked then a horizontal line as a yellow framed line in the color of the selected variable will be drawn at the average value of all plotted data of the **Selected Variable**.

If the **Selected Variable** is of the counter type, and these Poisson distributed data can be approximated by a Normal Distribution, two horizontal dashed lines will be drawn indicating the theoretical 95% range for the plotted data set, i.e. 95% of all data fall into this range, and 5% will be outside. If GeigerLog determines that the condition of Normal Distribution is **not** met, then **no** 95% range lines will be drawn, which is typically the case when the average is < 10 [9]).

If the checkbox **MvAvg** is checked then a Moving Average [10]) as a yellow framed line in the color of the **Selected Variable** will be plotted. The default duration for the moving average is 60 sec. E.g., with CPS data recorded once per second, applying a MvAvg of 60 sec will basically make a CPM curve out of it. For longer recording times moving averages over 600 or even 6000 may be appropriate.

GeigerLog will determine the average cycle time from the data. However, if the cycle time had been changed during the recording, this may not be adequate; adjust the duration entered to achieve a better fit.

8 http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=5369 see Reply #10, Quote: "With Thorium = 0.468, and K40 = 0.494, I'd finally put the calibration factor for the 2nd tube, the SI3BG, to 0.48 µSv/h/CPM. Which makes it 74 fold less sensitive than the M4011!"

9 For a more detailed discussion of Normal and Poisson Distributions of Geiger data see my "Potty Training for Your Geiger Counter" article on SourceForge https://SourceForge.net/projects/geigerlog .

10 The Moving Average, sometimes also called a Rolling Average is calculated and plotted by taking N data points, calculating their arithmetic average, and plotting the result at the time point in the middle of the range. Hence, N/2 data points at both the beginning and the end of the record will not be available in the Moving Average line.

The colored rectangle next to Avg (dark-blue in Fig. 4) shows the color the **Selected Variable** has in the plot. Clicking this rectangle opens the **Color Selector**, allowing you to select a different color for this variable. Clicking the Reset button, or reloading the database, sets the color back to the original.

Note: Sometimes, in particular when your log file contains very low and very high counts, it is advantageous to plot the data not in linear but in logarithmic scale. This can be achieved by clicking the Graph Toolbar icon labeled ‘Edit axis, curve and image parameters’ and then selecting Scale Log for the Counter-Y-axis.

You can also do further modifications via this toolbar icon, e.g. line color, line width, symbols and more.

The **SuSt**, **Stats**, **Poiss**, and **FFT** buttons are tools for the Quality Control of your data. The **SuSt** button prints a Summary Statistics of all variables currently displayed to the NotePad. The other three buttons act on the **Selected Variable** only and present their info in a pop-up-window. More on this in the next chapter Quality Control of your Data on page 23.

Display Last Log Values

The **Last Value Box** displays the last measured value of the **Selected Variable.** By clicking on this box a pop-up window shows the last values of all mapped variables and their device source. This window is auto-updated during logging.

When logging stops, the window remains open, but the values remain frozen and are shown on a gray background.

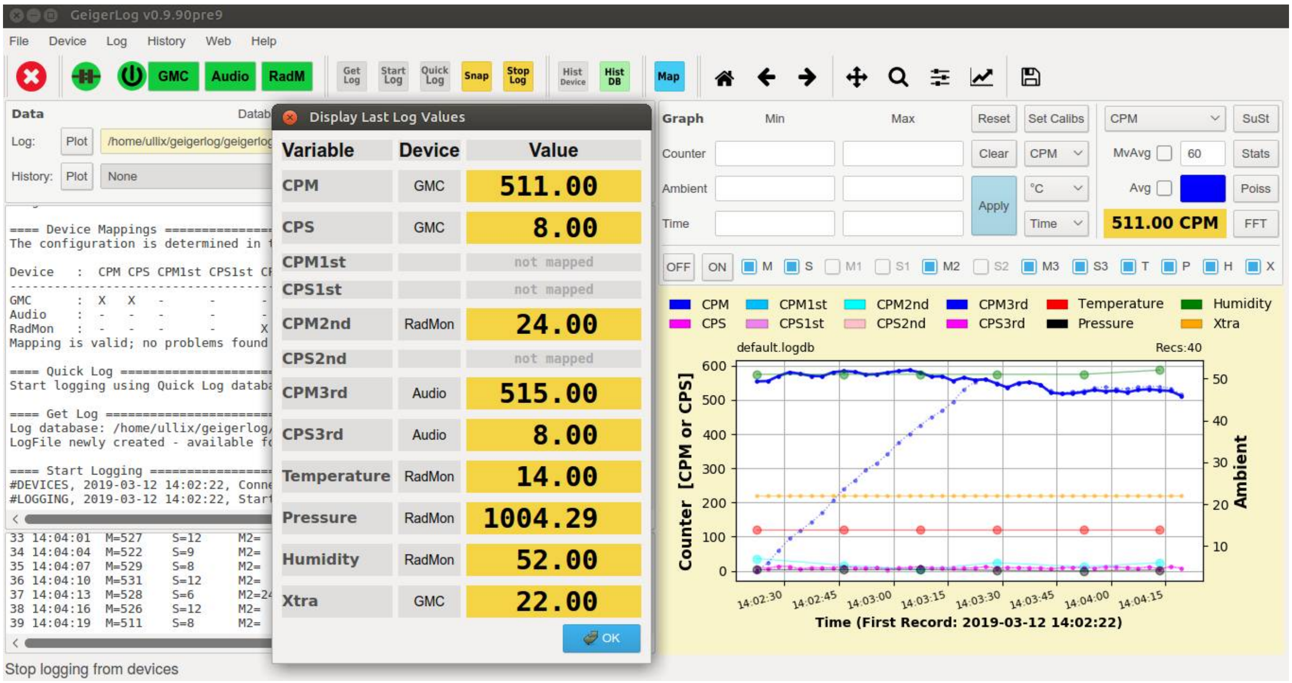


Fig. 6 The Display Last Log Values Pop-up Window

Configuration of GeigerLog

GeigerLog uses the configuration file `geigerlog.cfg`. This file is required, and at least one device must be activated in this file for GeigerLog to work! In the default state, a GMC device and an AudioCounter device are activated.

With the many different devices supported, some having model specific firmware bugs, changing firmware, new features and more, the config file may allow that an as yet unknown device can be configured so that it can be made to work with GeigerLog. Some of the firmware issues are laid out and explained in Appendix F – Firmware Differences on page 71.

A recent example is the calibration factor to transform values in CPM into µSv/h. It turned out that some of these factors as laid down in the firmware are wrong. Fortunately the user can easily configure them in GeigerLog's configuration file.

With GeigerLog supporting the WorldMap feature, the required information on the user and the device can be entered in the configuration file.

Scaling

GeigerLog has received a powerful formula interpreter to allow adjustments in support of environmental variables like temperature, air-pressure, and humidity, but this also allows dead-time correction of counter values.

Examples:

- The temperature may need adjustments, which can be as simple as:
  T_new = T_old – 0.23
- To plot pressure data – typically within 970 … 1030 hPa – on the same scale as other environmental parameters – Temperature -10 … 40 °C, Humidity 20 … 90% – the pressure could be plotted as:
  p_new = p_old – 1000
- The air-pressure is measured at the altitude of the location of your device, but typically for weather stations, you want it reduced to sea-level altitude. The formula is:
  P_sealevel = P_altitude * (1 - (0.0065 * altitude) / (20 + 0.0065 * altitude + 273.15))**(-5.257)
- You can apply a dead time correction for a Geiger counter. While I don't recommend doing it, because the counters have other problems that need to be solved first, if you want to use it the formula is :
  CPM_true = CPM_observed / (1 - CPM_observed * 200 * 1E-6 / 60))

There are two types of scaling : **ValueScaling** and **GraphScaling**.

ValueScaling: the measured value is modified and this modified value is saved. This is adequate if you know that your instrument is off by a certain amount and you correct this to save the proper value only.

GraphScaling: The original value is saved, and the scaled value is used only for plotting. The air-pressure example above is a typical application and implemented in the configuration file.

Loading *.log, *.his and Other CSV Files

The current version of GeigerLog stores its data in SQL-database files, while the previous versions had used CSV (Comma Separated Variables) text files. Log files had the extension ‘.log’, and History files the extension ‘.his’. Also, other programs may have created CSV files, which you might want to load into GeigerLog for analysis.

This is easily done with functions in the menu: **Log → Get Log from CSV File**, and **History → Get History from CSV File** [11]). You will be offered to load an existing *.log, *.his, or *.CSV file, which will then be presented in the Get Data from CSV File dialogue. An example is in the figure.

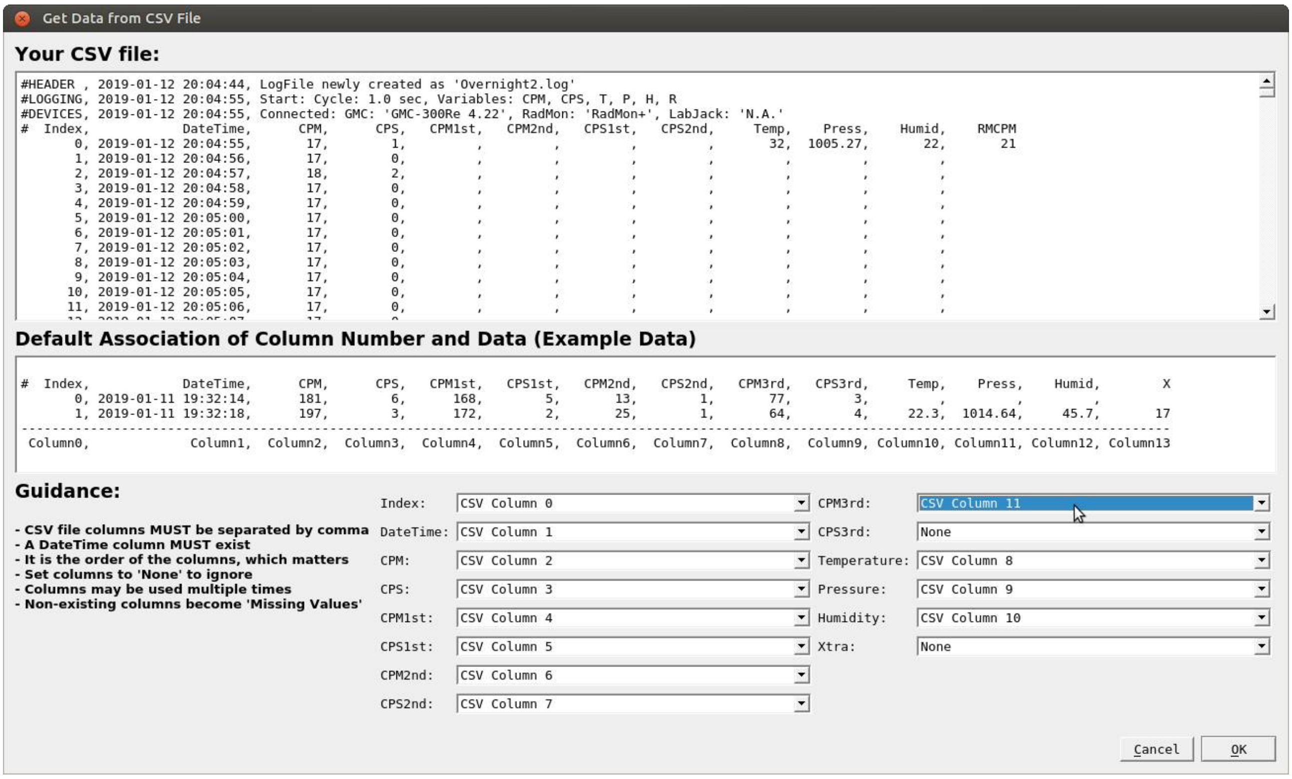


Fig. 7 The Get Data from CSV File dialogue

The top part shows a segment from the just loaded file, the middle part an example of a current mapping, and the bottom part allows to associate the columns of your CSV file with the variables of GeigerLog. The CSV file may have up to 20 data columns. One of them MUST have a Date&Time stamp format, and that MUST be associated with the GeigerLog variable DateTime. The other data columns can be associated freely with the variables.

In the example I had to “relocate” the RMCPM value to CPM3rd, spare CPS3rd, and shift Temp, Press, Humid to places upwards. Upon clicking OK a database file will be created, and you won’t have to redo this loading.

Saving data as *.log, *.his Files formatted as CSV

Using **Log → Save Log Data into *.log file (CSV)** and **History → Save History Data into *.his file (CSV)** will save the respective data as a CSV file with extension ‘log’, or ‘his’, resp.

11 The two actually aren’t different; it just helps to organize any data by their way of dreation

Software to handle database files

The software **DB Browser for SQLite** is Open source and available for download for Linux, Windows, and Mac, and is an excellent tool for SQLite3 databases: https://sqlitebrowser.org/ !

Quality Control of your Data

The **SuSt**, **Stats** , **Poiss**, and **FFT** buttons in the Graph Dashboard help you to check the quality of your data. In addition **Scatter Plot** and **Printing Values of the Variables** provide further help.

For each of these functions only the data currently shown in the plot will be included in the calculations! If you want to see the result for all the data in the file, click the **Reset** button in the Graph Dashboard first.

Also, a variable can only be used for analysis if that variable is shown in the plot. For the SuSt function all the variables in plot will be used. The other three functions use only one variable at a time, and it will be the selected variable, see Visualize Your Recordings – Graph Dashboard on page 16.

Furthermore, the variable values will be used in the units currently selected in the Graph Dashboard. CPM and CPS values may be shown in units of CPM or CPS, or of µSv/h. Temperature may be shown in °C or °F.

SuSt – Summary Statistics

Clicking **SuSt** will give a printout of some summary statistics in the NotePad. It may look like this:

```
==== Summary Statistics of Variables selected in Plot ============
File      = /home/ullix/geigerlog/geigerlog/data/default.log
Filesize  =    277,076 Bytes
Records   =        806 shown in Plot
          [Unit]      Avg  StdDev     Variance        Range          Last Value
CPM     : [CPM]     19.45 ±4.23          17.89       10 ... 33          6736.97
CPS     : [CPS]      0.33 ±0.559          0.31        0 ... 2            101.00
T       : [°C]      28.20 ±3.55e-15       0.00     28.2 ... 28.2          29.20
P       : [hPa]   1012.41 ±0.0753         0.01  1012.29 ... 1012.53     1011.67
H       : [%]       36.11 ±0.398          0.16       35 ... 37            35.00
X       : [x]       17.89 ±3.85          14.84        8 ... 26            22.00
```

As a first easy check for the validity of CPM and CPS values look at Average and Variance – they should be about the same (is the case here), unless you had varying conditions during a recording.

The reason for this lies in the properties of a Poisson Distribution, which is the relevant statistics for radioactive events. For an introduction to Poisson Distribution and its statistics see my “Potty Training for Your Geiger Counter” article available on SourceForge at https://SourceForge.net/projects/geigerlog/.

Note that this applies ONLY when the units CPM or CPS are used, and NEVER when µSv/h is used!

Likewise, for the variables temperature, air-pressure, and humidity the comparison of average and variance makes no sense!

Stats – Statistics

Clicking **Stats** will open a pop-up window showing standard statistics, which will have content like this:

```
==== Data as shown in the plot for selected variable: CPM =====================
from file: /home/ullix/geigerlog/geigerlog/data/default.log

Totals
  Filesize  =    311,786 Bytes
  Records   =        806

Variable: CPM (in units of: µSv/h)
                       % of avg
  Average   =    0.13      100%       Min  =    0.07        Max  =    0.21
  Variance  =    0.00     0.60%
  Std.Dev.  =    0.03    21.74%       LoLim=    0.10        HiLim=    0.15
  Sqrt(Avg) =    0.36   281.23%       LoLim=   -0.23        HiLim=    0.48
  Std.Err.  =    0.00     0.77%       LoLim=    0.13        HiLim=    0.13
  Median    =    0.12    97.68%       P_5% =    0.08        P_95%=    0.17
  95% Conf*)=    0.05    42.61%       LoLim=    0.07        HiLim=    0.18

  *) Approx. valid for a Poisson Distribution when Average > 10

Time
  Oldest rec    = 2018-08-19 13:49:41  (time=0 d)
  Youngest rec  = 2018-08-19 14:29:56  (time=0.028 d)
  Duration      = 2415 s   =40.25 m   =0.6708 h   =0.02795 d
  Cycle average = 3.00 s

First and last 7 records:
#HEADER , using Quick Log file: default.log
#LOGGING, 2018-08-19 13:41:50, Start with logycle: 3.0 sec
#LOGGING, 2018-08-19 13:41:50, Log variables: CPM, CPS, T, P, H, R
#LOGGING, 2018-08-19 13:41:50, Connected GMC Device: 'GMC-300Re 4.22'
#LOGGING, 2018-08-19 13:41:50, Connected RadMon Device: 'RadMon+'
#Index,           DateTime,      CPM,      CPS, Temp,   Press, Humid,    RMCPM
    0, 2018-08-19 13:41:50,       16,        1, 27.2, 1012.52,    36,       14
...
 2388, 2018-08-19 15:41:14,  6778.79,    121.9,
 2389, 2018-08-19 15:41:17,  6761.01,   116.66,
 2390, 2018-08-19 15:41:20,  6773.56,   110.38,
 2391, 2018-08-19 15:41:23,  6794.47,   109.34,
 2392, 2018-08-19 15:41:26,  6803.89,    96.84,
 2393, 2018-08-19 15:41:29,   6801.8,   127.15, 29.2, 1011.5,     35,       22
 2394, 2018-08-19 15:41:32,  6796.57,   114.57,
```

Poiss – Histogram with Poisson Fit

This tool is relevant ONLY to Geiger counter data shown in CPM or CPS, but for these it is immensely useful!

It does NOT make sense to use it when Geiger counter data are shown in µSv/h (or any other dose rate, like mR/h or else). It also does NOT make sense to use for environmental data, like temperature, air-pressure, humidity, as none of these have an underlying Poisson distribution!

The next two figures provide examples of histograms with a Poisson fit; fig 5 for low count rates as in a background measurement, and fig. 6 for a much higher count rate.

The value r² (in the graph as r2) is an indicator for the goodness of a fit. **A value of r² ≥ 0.9 suggests a proper measurement.** If r² is smaller, then there may not be enough data points for a meaningful average, or some experimental error (source or counter shifted or removed during data collection ?) may have occurred.

Use the Poisson Test as an essential quality control tool for your measurement.

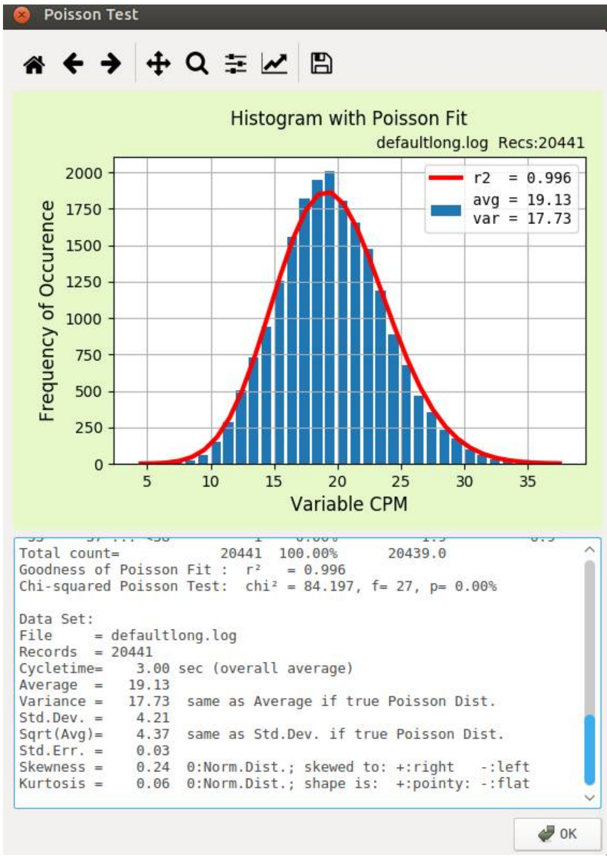


Fig. 8 Histogram of Low Count Rate

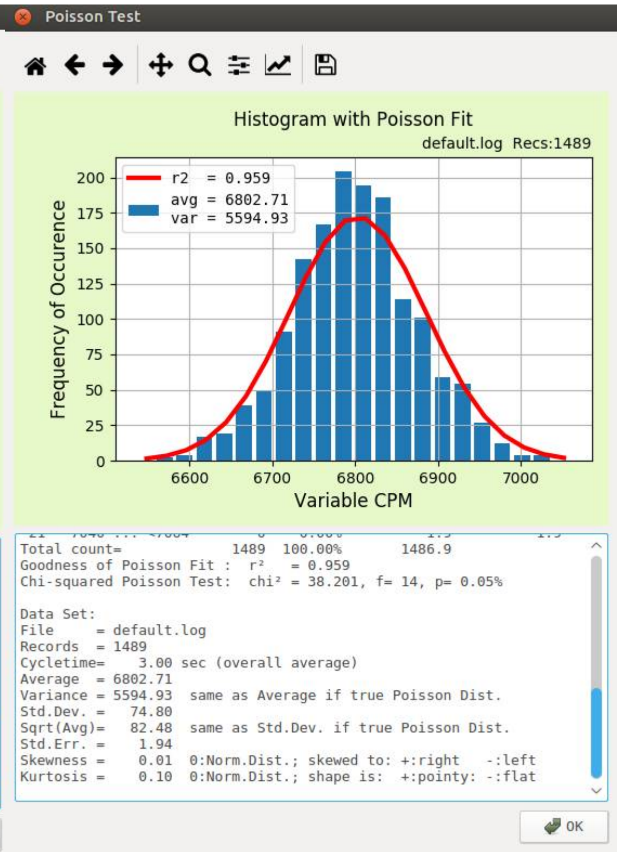


Fig. 9 Histogram of High Count Rate

FFT – FFT & Autocorrelation Analysis

The FFT (Fast Fourier Transform) allows to analyze a time dependent signal, like the Count Rate, for any periodic signal hidden within the data. An example is given in fig. 7. The data were recorded by logging in the CPM mode.

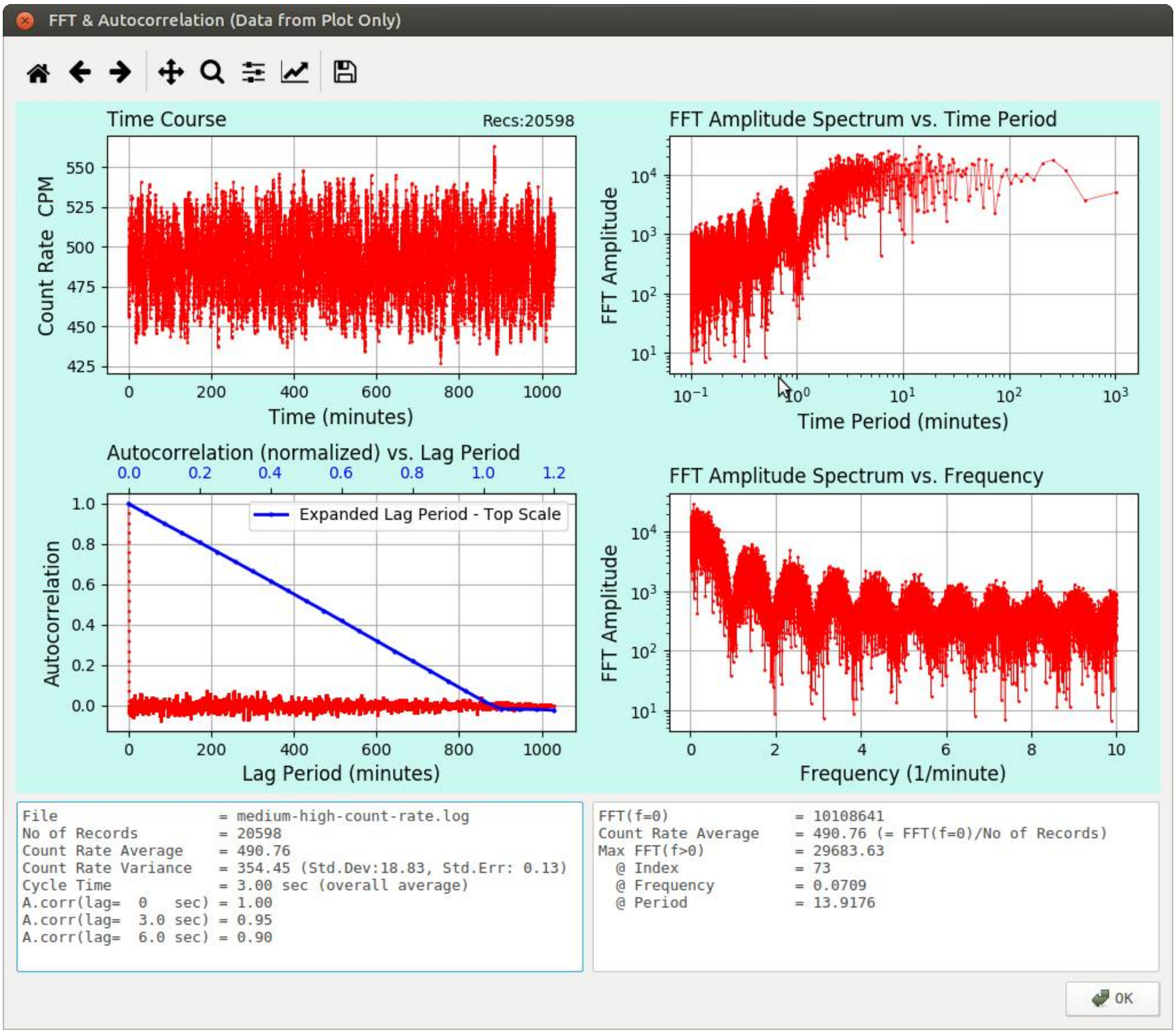


Fig. 10 FFT Analysis of Medium Count Rate Measurement

With a bit of squinting at the Time Course of Count Rate vs. time (upper left panel) one may expect to find a signal with a period of 1 or more hours; at least I did. However, in the range of (upper right panel) >1 to 1000 minutes there is no such signal. Instead there is a very pronounced signal at a period of 1 min, equivalent to a frequency (bottom right panel) of 1/minute. This frequency plot clearly also shows all the harmonics of this frequency.

The effect is independent of count rate (same pattern at background count rates) and sampling time (< 30sec). At a sampling time of > 30sec, this signal would not be observable anyway due to the Nyquist limit.

The fact of a pronounced 1 **min** Period in the FFT spectrum, and the Counts per **Minute** sampling, raised the suspicion, that this was related. But, as was first considered, it has nothing to do with the Geiger counter taking a little break every minute. Rather, it is the consequence of oversampling.

In this experiment the CPM readings were taken every 3 seconds. CPM is the sum of readings during the last 60 seconds. The next reading 3s later has 3 “fresh” seconds of data, and has dropped 3 “old” seconds of data. But 57s worth of data remain unchanged. Which means that all data taken over 60s are related, strongly initially, and weakly at the end.

Such a relationship can be quantified by calculating the autocorrelation of a signal. This is shown in the bottom left diagram of fig.7. The data are redrawn in blue vs. an expanded Lag Period (labeled on top of this panel). And, indeed, one sees the autocorrelation dropping linearly from the initial 1 (highly correlated) to the 0 (= non-correlated) at exactly 1 minute.

So, it is autocorrelated, what does it have to do with the FFT spectrum? The autocorrelation can be seen as the convolution (or folding, different name for the same thing) of a rectangle in time of length 1 minute and a Poison distribution of the Geiger data. The FFT spectrum is then a mix of the rectangle spectrum and the Poisson spectrum.

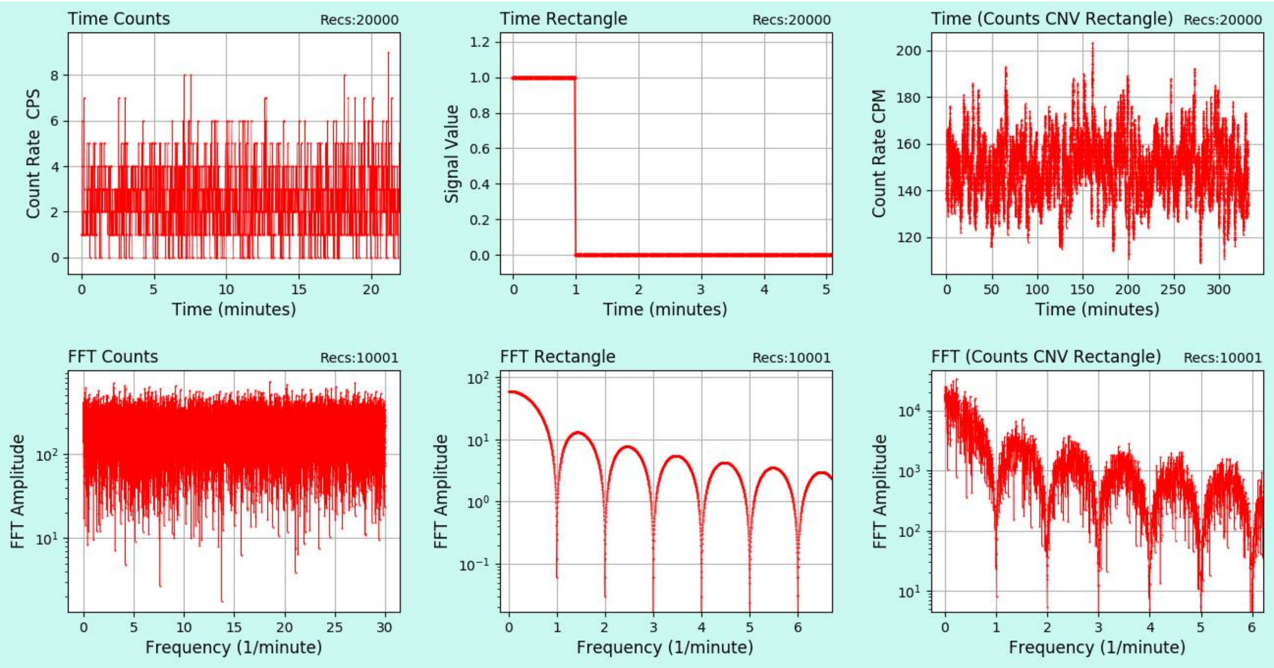


Fig. 11 Demonstration of the impact of convolution on an FFT spectrum (Synthetic data, CPS=2.5)

This can be nicely demonstrated using synthetic recording. In fig. 8 the upper panels show the signals in the time domain from Poisson White noise at average CPS=2.5 (upper left), a rectangle of 1 min at value 1 and value 0 for the remaining 19988 counts (upper middle), and the convolution of these two signals (upper right), resulting in average CPM=150. The bottom panels show the corresponding FFT spectra, white noise, a 1/min frequency and harmonics, and the mix of the two. Oversampling does no harm; but it must be accounted for when autocorrelation plays a role.

Printing Values of the Variables

Sometimes you want the numerical values of your variables. Use the command in menu **File → Print Data as Shown in Plot** to print the Date&Time and the values of variables into the NotePad, but print only those variables currently shown in the plot, and only for the time frame selected in the plot. Makes it easy to inspect values in a limited range. You can save this printout to a file using menu **File → Save NotePad to File**, or print it on paper or as a pdf file **File → Print NotePad.**

Scatter Plot

It is sometimes helpful to plot one variable against another one in an X-Y-scatter plot. You might be wondering whether temperature is correlated with humidity? Or the air-pressure has an influence on the temperature?

Another question was of particular relevance for this version of GeigerLog: given that we can measure Geiger counts with one single GMC-device by both the technique of digitally transmitting the data via the USB cable, and at the same time by the audio cable, the two measurements should be strongly correlated. Are they? I use the data referred to in the chapter AudioCounter Devices on page 35.

Use the command in menu **File → Plot Scatter**. In the upcoming dialog you choose the variables for the X- and Y-axis, which in this example are CPM (digital data), and CPM3rd (audio data).

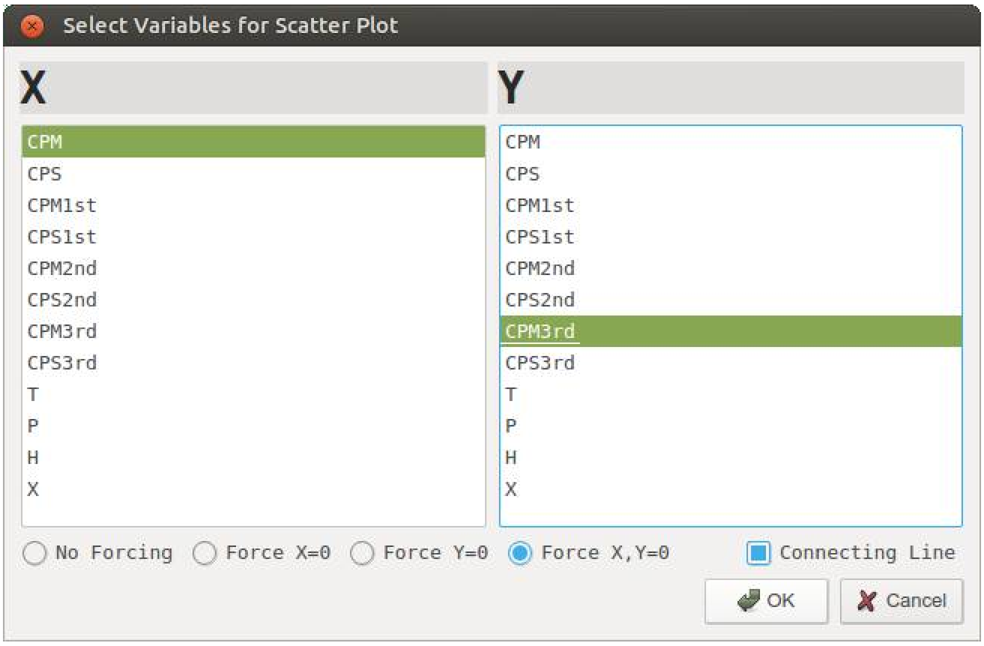
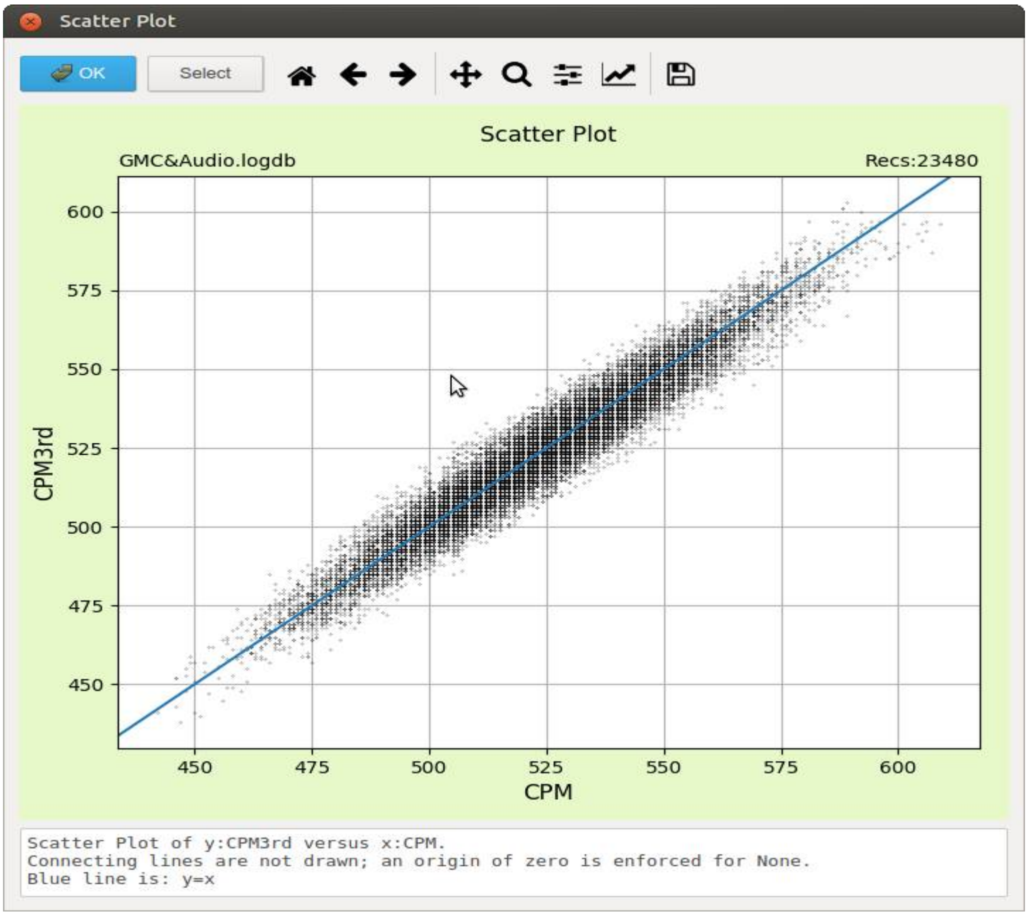
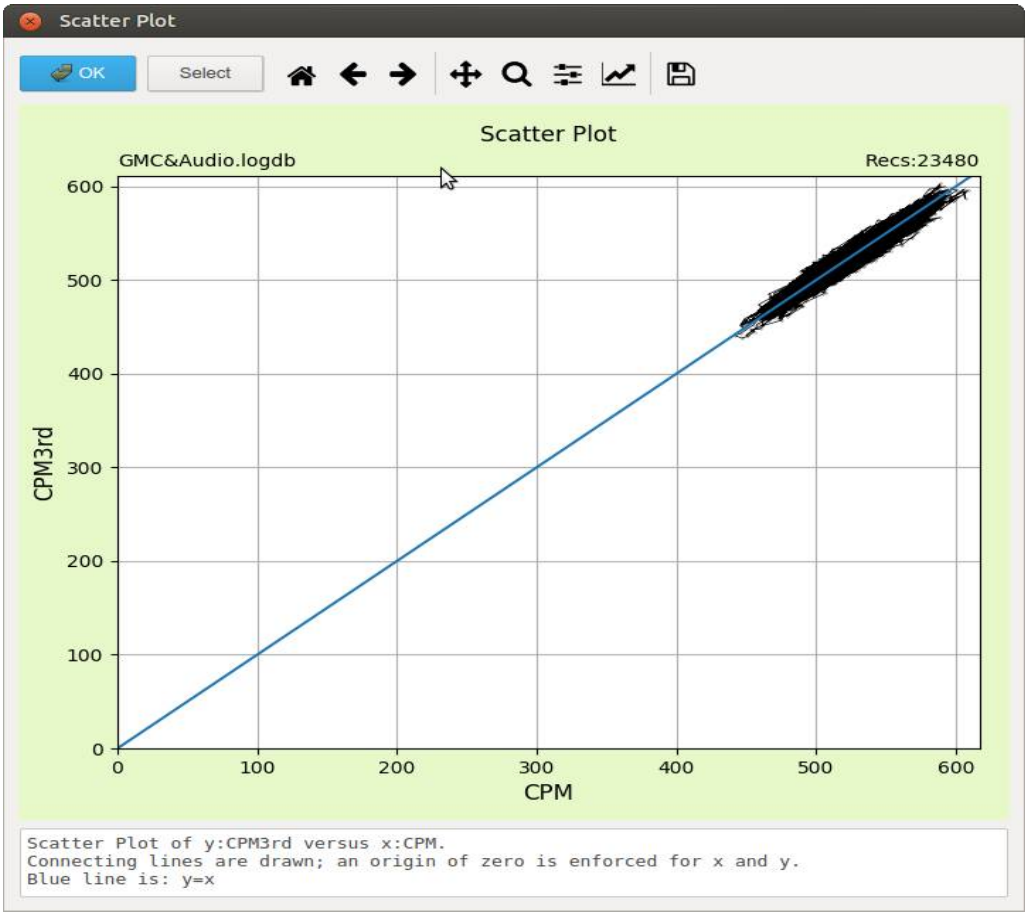


Fig. 12 Select Variable dialog for the Plot Scatter command

After clicking OK, a pop-up-window shows the scatter-plot, demonstrating the clear correlation between the two variables. The blue line is y=x; it helps to judge the quality of correlation of the two variables plotted – it is excellent in these graphs!

Note that you can force to include the origin x=0, y=0 in the graph, or only x=0, or only y=0, or neither, and that you can have the data points connected with lines, or not. Click button **Select** to restart the selection dialog.


Scatter Plot
OK
Select
Scatter Plot
GMC&Audio.logdb
Recs:23480
CPM3rd
CPM
Scatter Plot of y:CPM3rd versus x:CPM.
Connecting lines are not drawn; an origin of zero is enforced for None.
Blue line is: y=x


Scatter Plot
OK
Select
Scatter Plot
GMC&Audio.logdb
Recs:23480
CPM3rd
CPM
Scatter Plot of y:CPM3rd versus x:CPM.
Connecting lines are drawn; an origin of zero is enforced for x and y.
Blue line is: y=x

Device-specific Considerations

GMC-Devices

Operating Modes

GMC-Devices support the operating modes **Logging** and **History**.

For **Logging** a connection to GeigerLog must have been established, and the device must be powered-on. Then it is GeigerLog initiating and executing all communication and data transfer to and from the device.

History is a stand alone operation of the device. It must be powered-on, and then collects data and stores them in its internal memory controlled by its own microprocessor. It may remain connected to a computer and GeigerLog, but it does not have to.

However, in order to read the data from the device, it needs to be connected to GeigerLog. GeigerLog will initiate and execute the data transfer from the device. The data can then be handled in GeigerLog as if they were a logging recording.

General: GMC-device functions like Speaker, Alarm, Saving mode, Date&Time, Calibration, Threshold, can be read by GeigerLog, and some can also be set. All communication with the device is error-checked and corrected if possible.

Connecting

The GMC-Devices require a USB cable connection. Unfortunately, this is not a modern USB connection, but an old-style serial connection, electrically handled via a USB-to-Serial converter, which is built into the GMC-device. This has the consequence that the computer cannot recognize the connected device (it only recognizes the converter)! And this has the consequence that establishing a connection to a GMC-device may be more challenging than expected.

The two essential parts for a connection are: 1) the **USB-Port name**, and 2) the **Serial Baudrate**. GeigerLog may be able to auto-discover the GMC-device by exercising some tests on the connection, called the USB Autodiscovery, available in menu **Help → Autodiscover connected USB Port**. However, observe this Warning:

WARNING: If you have other devices – besides any GMC Geiger counter – connected with a **USB-to-Serial** adapter (which is what your counter has built-in) it is possible that the other devices are significantly disturbed by the USB Autodiscovery!

It is recommended to halt or switch off those other devices BUT LEAVE THEIR USB CABLE CONNECTED, and only then start the USB Autodiscovery.

Any other native USB devices (mouse, keyboard, printer, ...) will NOT be impacted by the test.

When your USB-Autodiscovery ended successfully a dialog box will pop-up and tell you that it found a GMC-device and let you click the OK button to use it right away. However, you might want to make this more permanent by editing the configuration file `geigerlog.cfg` in its section **GMCSerialPort** to make these just found settings the default.

Before going into more details, give it a try first. Start GeigerLog, and select menu **Device → Connect Device** as explained in Establishing a GeigerLog Connection to Your Devices on page 11.

If unsuccessful, a printout **in red** will tell you the reason. You will likely be advised to run **USB Autodiscovery** from menu **Help**. For more complicated situations see Appendix B – Connecting GMC Device and Computer on page 59.

If successful, you'll see the GMC icon turn green, and a **GMC Device Info** text is printed to the NotePad. You're set to go.

Your GMC Geiger Counter Model

GeigerLog works the same for all GMC-Devices except for some workarounds accounting for the different firmware, firmware bugs, memory sizes, calibration factors and more. It therefore is important that after you have made the connection the correct Geiger counter model and firmware is shown in the printout to the NotePad!

If this is not the case, then you may have to customize your model by modifying the configuration file `geigerlog.cfg` in its **GMCDevice** section. Some of past problems are highlighted in Appendix F – Firmware Differences on page 71.

It is now assumed that a successful connection of the GeigerLog with the Geiger counter has been established.

Powering On

For a working connection between computer and Geiger counter, the counter does not have to be switched on (powered on); it can remain off. The power for its electronics comes from the USB port, thereby also charging the battery. In this mode you can read and set various parameters of the counter, and you can download the history.

But for all new radiation measurements – be it by Logging or by History – the Geiger counter must be powered on. This power switching can be done manually directly at the device, or easier from GeigerLog (menu **Device → GMC Series → Switch Power ON**). GeigerLog's GMC device power icon

will change its state from Power OFF  to Power ON . When the icon is gray then GeigerLog has not yet been able to determine the power state of the counter.

Note that you can easily toggle the power state by clicking this icon on the toolbar!

Logging

Any logging is strictly controlled by GeigerLog, **not** by the counter [12])!

12 This is different from the way GQ's Dataviewer software works. DV uses the outdated heartbeat function of the counter, which only provides CPS readings, and does not allow any simultaneous other communication with the counter. Thus it is impossible to use any of the more recently introduced functions for reading more than a single tube.

For every value GeigerLog wants to have, it must send a specific command to the counter. The counter answers with the data. After GeigerLog has obtained all values for one cycle, it saves them as one record, prints them to the LogPad, and displays them in the graph. Then it waits for the cycle time to expire to start asking for the next record of data.

The values always asked for by GeigerLog are:

CPM	: Counts Per Minute
CPS	: Counts Per Second

Since the release of the GMC-500+ counter, which has not just one but two Geiger tubes installed, its firmware was extended to allow reading the tubes individually. For this device the values asked for by GeigerLog are:

CPM	: Counts Per Minute as the sum of both tubes (makes no sense [13]))
CPS	: Counts Per Second as the sum of both tubes (makes no sense)
CPM1st	: Counts Per Minute for the 1st tube, the standard tube
CPM2nd	: Counts Per Minute for the 2nd tube, the low-sensitivity tube
CPS1st	: Counts Per Second for the 1st tube, the standard tube
CPS2nd	: Counts Per Second for the 2nd tube, the low-sensitivity tube

These commands work error-free on all counters, also those with single tubes only as well as with older firmware, but on all devices, except the GMC-500+, the answers are redundant:

CPM = CPM1st = CPM2nd
CPS = CPS1st = CPS2nd

An example of a Logging with a GMC-device in combination with an AudioCounter as a 2nd device is shown in Fig. 13 in chapter AudioCounter Devices.

History

Any GMC Geiger counter can measure the counts from radiation and store the results in its internal memory, not needing a computer connection. In the older units this memory size is 64kB (65536 bytes). For a CPS measurement, this suffices for almost one full day of measurements. For a CPM measurement the memory would last roughly from 1 to 5 weeks. The duration depends strongly on the intensity of the radiation due to the storage algorithm implemented in the Geiger counter firmware. It should easily cover even an extended vacation, unless you plan on camping inside a damaged nuclear reactor!

Newer units have an internal memory of 1MB, extending the collection spans even further.

However, this is not necessarily an advantage. Downloading just the 64K already takes about 25 sec at the fastest serial speed! Downloading 1MB takes ~5min. This is where a faster speed would really be helpful.

In theory you could download only a portion of the memory. But since this is laid out as a ring-buffer, you'd have to know very precisely what portion of the memory you want. Typically you won't know this until after you have done the complete download and inspected the data. On top of this, a

13 See discussion e.g. here: http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=5304

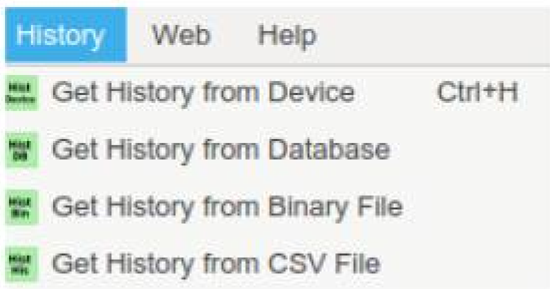
partial download may bring parsing problems (see Appendix E – GMC Device: Internal Memory, Storage Format and Parsing Strategy). So this is not an option.

As a 5 min download is really inconvenient, the GeigerLog protocol for the download has been modified to: the download will be stopped when 8192 bytes, each having hex value ‘FF’, have been read. Unfortunately, ‘FF’ can both be a legal value for CPM or CPS, but also signals erased or empty memory. As the memory is organized in pages of 4k bytes each, it means that 2 successive pages of ‘FF’ must be found. This can only be the case when nothing is written into these 2 pages nor into any pages beyond them. So we can safely stop downloading.

However, if the memory overflows, the ring-buffer (see Appendix E – GMC Device: Internal Memory, Storage Format and Parsing Strategy) storage principle becomes effective, and the memory is overwritten beginning at the bottom. In this situation the whole memory is filled with data, and there will never be 2 pages of empty values. Hence the whole memory will be read!

If you don’t need the content of the memory, I suggest to erase it every once in a while. Unfortunately, on the older counters this can be done with a Factory Reset only. Some newer counters provide a separate command to erase the memory.

NOTE: if you experience reading errors while downloading the history, or even a partially or completely unreadable data, try to increase the timeout setting in the **GMCSerialPort** configuration section of the configuration file `geigerlog.cfg`!



Handling the History is controlled by two buttons in the toolbar, and the commands available in Menu – History.

The toolbar button **Hist Device** downloads the history from the GMC-device to your computer, creates a database file, and plots the data. With ongoing Logging this button is inactive, as the GMC-Devices stumble with parallel logging and downloading!

The buttons **Hist DB** load data from a database files created by a previous history download, see more at Menu – History.

The menu History offers two additional commands: **History → Get History from Binary File** reads a binary history file created by an earlier version of GeigerLog, or by a different software, parses the data, and creates a regular database file. **History → Get History from CSV File** reads an already parsed history file from a previous version of GeigerLog – then called *.his file – or a different software producing files in CSV format.

A History example – data collected during a long distance flight – is shown in Fig. 1.

History Saving Mode

The GMC counters can use different strategies to store the data in the history memory, ranging from not storing at all, to storing CPS or CPM in different time intervals, or even conditional on exceeding a count threshold.

The mode can be switched using the **Mode** button in the **Timings** column of the **Data Dashboard**. The button shows the abbreviations for the different modes:

Mode: OFF	- OFF (no history saving)
Mode: CPS	- CPS, save every second
Mode: CPM	- CPM, save every minute
Mode: CPMh	- CPM, save hourly average
Mode: CPSTh	- CPS, save every second if exceeding threshold
Mode: CPMTh	- CPM, save every minute if exceeding threshold

I strongly discourage using the threshold modes, as they distort the data and may make interpretation difficult or impossible, because you loose all knowledge on Poisson properties of the data!

Assembly

No assembly needed. Just a USB 2.0 A-Male to Mini-B Cable is needed like this: https://www.amazon.co.uk/AmazonBasics-Male-Mini-B-Cable-Feet/dp/B00NH11N5A

Configuration

GeigerLog auto-detects the type of connected GMC-device and adjusts itself to match features, and correct deficiencies and any known firmware bugs of the connected device.

However, sometimes GQ releases a new device or new firmware, without disclosing even essential changes. In those situations you may have to study the many settings in the configuration file `geigerlog.cfg` and make adjustments. Chances are good that you can make even a new device work. It may take some effort, though.

A ‘Factory-Reset’ is recommended to be sure of a defined starting condition. All settings relevant to GeigerLog can be set from within GeigerLog.

The exception is the baudrate used in the USB-to-Serial converter. This can only be changed at the device itself. I found the factory set baudrate of 115200 working well on a GMC-500+, whereas on a GMC-300E+ a baudrate of 115200 produced more hiccups in communication than its default of 57600.

AudioCounter Devices

Some Geiger counters generate audio-clicks for each registered radioactive event. For some counters – especially very old ones and modern low-cost varieties – this is the only means of indicating an event.

But even some very advanced designs of counters offer only an audio output, like in particular this recently reviewed, semiconductor based radioactivity detector **Smart Geiger Pro (SGP-001)** [14]).

GeigerLog now fully support those audio counters. How to operate, connect, configure and run such devices is described in the recently published article **GeigerLog-AudioCounter-Support-v2.0.pdf** available on Sourceforge: https://sourceforge.net/projects/geigerlog/files/Articles/GeigerLog-Audio-Counter-Support-v2.0.pdf/download

The article also contains a comparison of the digital measurement with the audio measurement using a Geiger counter, which can do both simultaneously, a GMC-300E+. The results fully confirm the validity of the two different methods. Fig. 13 shows an example of a simultaneous digital and audio recording with two different devices, a GMC-Device (GMC-300E+) connected digitally, and the **Smart Geiger Pro (SGP-001)** device, connected as an AudioCounter.

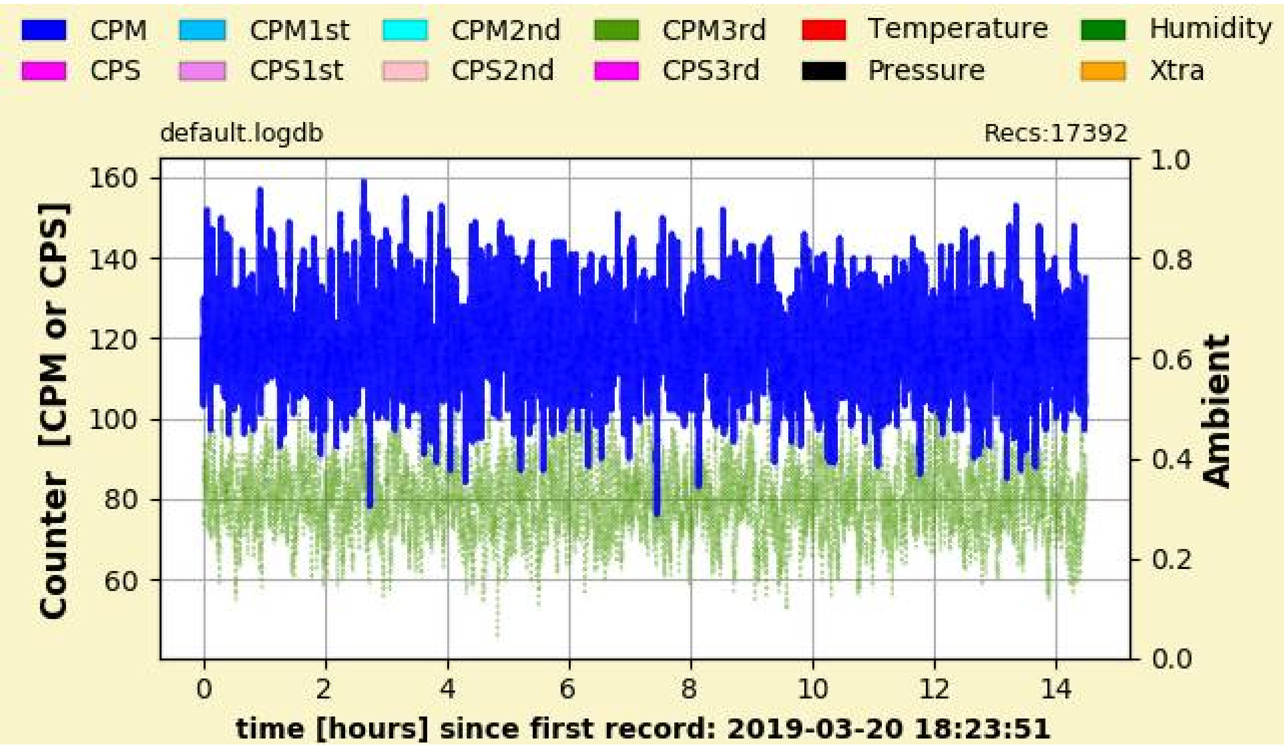


Fig 13: GMC-300E+ (blue) digital recording, and SGP-001 Device (green) audio recording

14 Read article **GeigerLog-Review Smart Geiger Pro (SGP-001)** on Souceforge at https://sourceforge.net/projects/geigerlog/files/Articles/GeigerLog-Review%20Smart%20Geiger%20Pro%20%28SGP-001%29-v.1.0.pdf/download

I2C Devices

GeigerLog can now handle I2C based sensors. Such sensors require an I2C connection, which is not available on today’s computers, at least not for regular users. However, USB dongles are available, which provide this type of connection.

I have tested 3 different dongles with several I2C sensors, and created Python based software to handle them. This software is available for download at the I2Cpytools project at Sourceforge (https://sourceforge.net/projects/i2cpytools/, [15]).

GeigerLog now implements a subset of the devices evaluated; presently implemented is the use of the ELV USB-I2C dongle from ELV Elektronik AG connected with a BOSCH BME280 sensor (temperature, air-pressure (i.e. atmospheric-pressure), and humidity) and the TSL2591 light sensor.

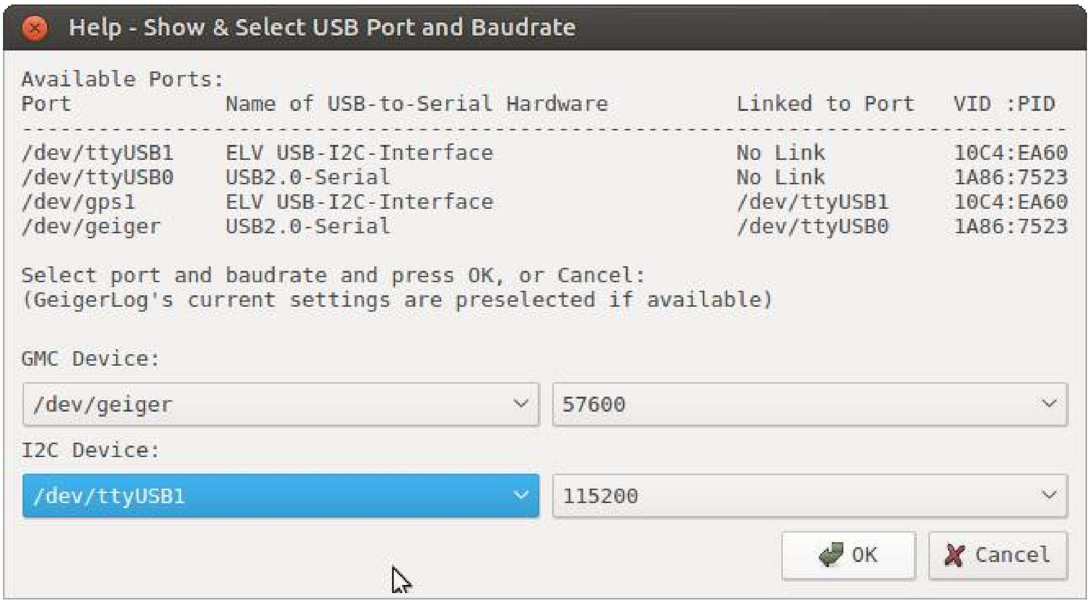
Find details on all devices in the **I2Cpytools – Manual (version 0.2)** [16]**)** on Sourceforge.

Connecting

The ELV dongle requires a USB cable connection. But just like with the GMC devices, ELV uses a USB-to-Serial converter, which results in similar difficulties for making a connection.

Again the two essential parts for a connection are: 1) the **USB-Port name**, and 2) the **Serial Baudrate**. Both are configured in the configuration file `geigerlog.cfg` under the heading **I2CSerialPort**. The baudrate can always be left at 115200, but the USB-Port must be given.

GeigerLog can assist you in finding the right USB-Port when you call menu **Help → Show & Select USB-Port and Baudrate**. This will present this dialog box:



You see that the ELV dongle is found at port /dev/ttyUSB1 on my Linux computer, which you can select in the drop-down box at the bottom, left.

On a Windows computer the port name will be COM3, COM12, or similar.

If the wrong port name is selected, and an I2C counter is activated, then establishing a connection might result is a similar problem as when running the USB-Autodiscovery for a GMC counter:

15 The most recent version is https://sourceforge.net/projects/i2cpytools/files/I2Cpytools-v0.2.zip/download

16 https://sourceforge.net/projects/i2cpytools/files/I2Cpytools-Manual-v0.2.pdf/download

WARNING: If you have other devices connected with a **USB-to-Serial** adapter it is possible that the other devices are significantly disturbed by the attempt to connect an I2C counter!

It is recommended to halt or switch off those other devices BUT LEAVE THEIR USB CABLE CONNECTED, and only then initiate the connection.

Any other native USB devices (mouse, keyboard, printer, ...) will NOT be impacted by the test.

Fig. 14 shows the last few weeks of a Long-Term – more than a year – recording using the ELV dongle and two sensors BOSCH BME280 (Temperature, Pressure, Humidity) and TSL2591 (light). The pressure is plotted as value[hPa] – 1000, Temperature in °C, Humidity as %relative, the light in arbitrary units.

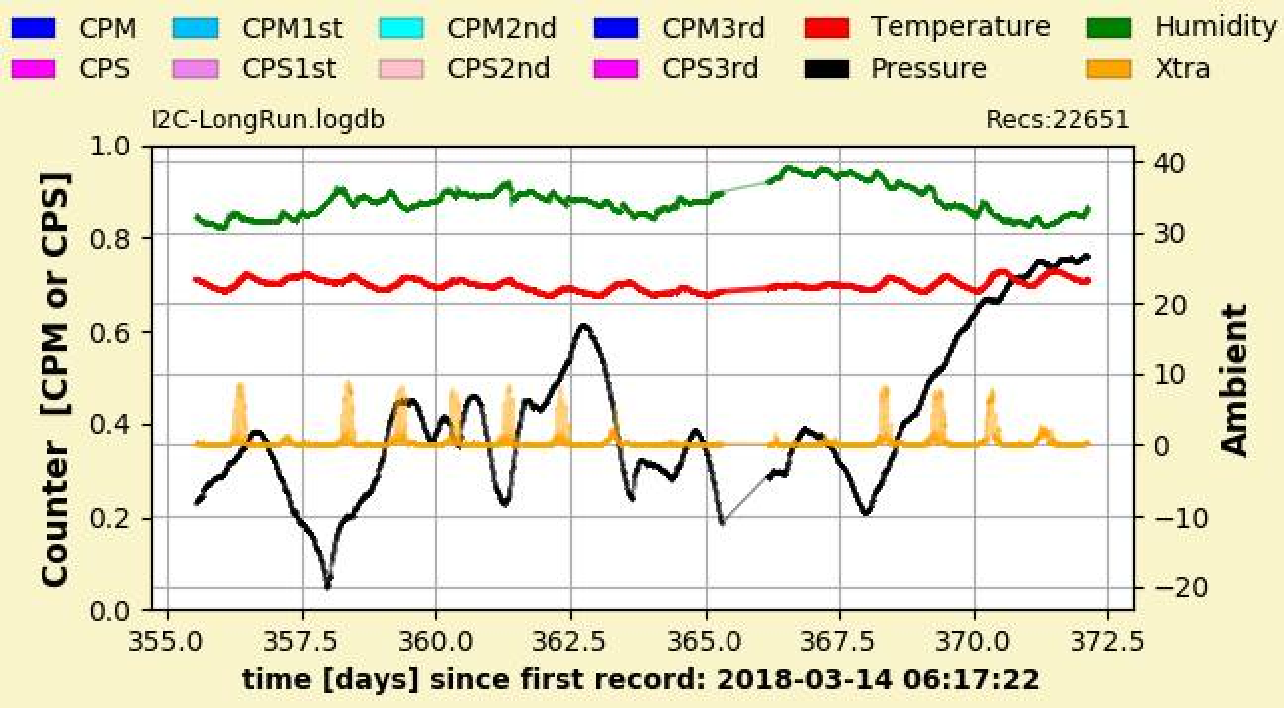


Fig. 14 A Long-Term recording Using I2C-Devices

RadMon Devices

The RadMon devices [17]) are part of a small family of devices, of which the **RadMon+** is used here. They come as Do-It-Yourself kits; assembly is required (see below).

Operating Modes

RadMon devices acts as IoT (Internet of Things [18]) devices, and send their data wirelessly to a special IoT server. GeigerLog reads the data from that server.

You have the option of installing on the RadMon either a Geiger tube, or an environmental sensor [19]) for temperature, air-pressure, and humidity, or both. GeigerLog can handle up to all four variables. In the default setting it is assumed that the RadMon has both a working Geiger tube as well as a working environmental sensor.

The RadMon device needs to be software configured at the device itself.

A demonstration mode can be activated in GeigerLog, which defines my personal RadMon+ device as active, and allows any user of GeigerLog to read genuine real-time data from a RadMon+.

Connecting

There will actually never be a connection between a RadMon device and GeigerLog, nor will the two ever talk to each other! That is the norm for IoT devices.

The RadMon will be configured to send its data to an IoT server – in IoT lingo a broker [20]) – into a specific folder whenever it has data ready. GeigerLog is told the name of the broker and the folder, and connects to the broker and tells him that it wants these data. The broker informs GeigerLog when new data are available, and GeigerLog downloads them.

To the user it looks like the two are connected, though technically they aren't.

Logging

As explained above, GeigerLog cannot ask the RadMon for new data, so you must configure the RadMon and GeigerLog independently.

First the RadMon is configured, e.g. by using a smartphone. Since I am making my personal RadMon available to GeigerLog users for a demo, I describe its configuration:

My RadMon+ is set to collect counts from the Geiger tube for 60 seconds to determine a CPM value. Then the RadMon+ reads the data for temperature, air-pressure, and humidity from its BOSCH BME280 sensor. All 4 values are then sent through my wireless home network to my router and then to a broker server located in North America [21]). The total of reading the 3 environmental

17 DIYGeigerCounter https://sites.google.com/site/diygeigercounter/
18 https://en.wikipedia.org/wiki/Internet_of_things
19 BOSCH BME280
20 A broker will be a server in your local LAN or anywhere on the internet, which runs MQTT software, https://en.wikipedia.org/wiki/MQTT . Tested servers are based on the Eclipse Mosquitto Open Source message broker Mosquitto https://mosquitto.org/ . You can easily install one on your own computer.
21 Initially I used the server **iot.eclipse.org**, but it seems to have been switched off. More recently I used: **broker.hivemq.com**. However, many more such servers are publicly available for testing purposes:

variables, and processing and shipping all 4 variables takes an extra time of about 7 seconds. Then the RadMon+ starts a new cycle.

GeigerLog is configured to connect to the same broker, is told what data to expect and where to find them on the server, but otherwise knows nothing about the RadMon+ device. The two sit only a few meters apart, but communicate via a 20000 km round trip of some typically 120 ms duration. A true variant of remote sensing ;-).

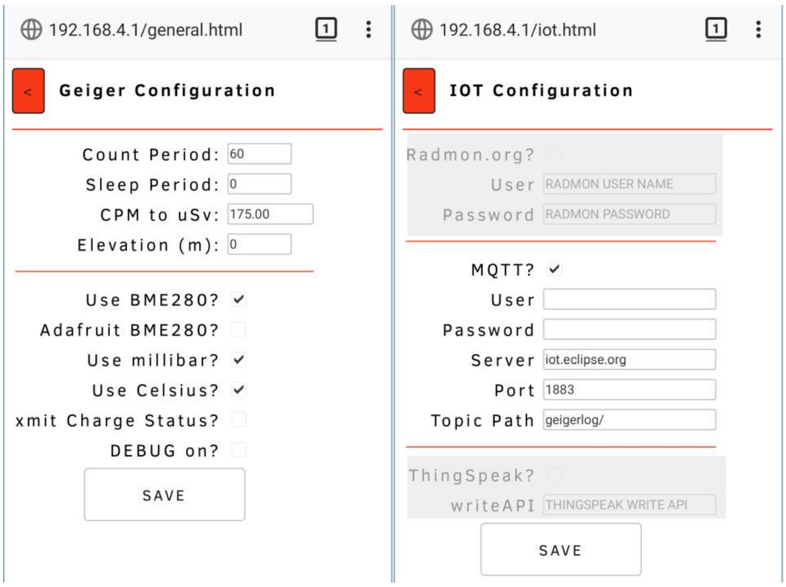
Obviously, both RadMon+ and GeigerLog must have WLAN/network and internet access.

Any GeigerLog user can configure his copy of GeigerLog to access my own RadMon+ device by activating it in the configuration file [22]). Upon establishing a connection, the device should be available. However, I cannot guarantee that my RadMon+ will be always on, but there is a good chance for it.

Assembly

The RadMon+ devices come as Do-It-Yourself kits; all parts – except the tube, power supply, and a case – are delivered, but you have to solder it yourself. Some basic skill in soldering is needed, but it is not overly difficult as there are no tiny SMD parts. You definitively want to have the manual [23]) ready when you do the assembly!

Configuration



RadMon Configuration

After assembly, the RadMon+ needs to be configured as explained in its manual. I found it easier to do this using my smartphone than my computer. The first page needs info on your own wireless LAN, so that RadMon+ can connect to it. The next two pages are shown in the figure, as screen shots from my smartphone:

The left page shows my current setting for the hardware, the right page for the IoT configuration; only the MQTT part is relevant.

Note that the 'Topic Path' is 'geigerlog/' – ending with a slash '/'! The same must be entered into GeigerLog's configuration file (see below) as 'RMServerFolder = geigerlog/'!

https://github.com/mqtt/mqtt.github.io/wiki/public_brokers

22 Simply set: RMActivation = yes in file `geigerlog.cfg`.

23 https://www.dropbox.com/s/ypmfjw97b8qlhs1/GK%20Radmon%20Build%20and%20User%20Guide%20v2.1.pdf?dl=1

Fig. 15 shows the last few weeks of another Long-Term –some 9 months – recording using the RadMon+ device with its sensor BOSCH BME280 (Temperature, Pressure, Humidity) and a SBM20 Geiger tube. The pressure is plotted as value[hPa] – 1000, Temperature in °C, Humidity as %relative, the Geiger data are blue scatter in the background. The device is located outside in a weatherproof housing. The CPM average is 20.1 CPM.

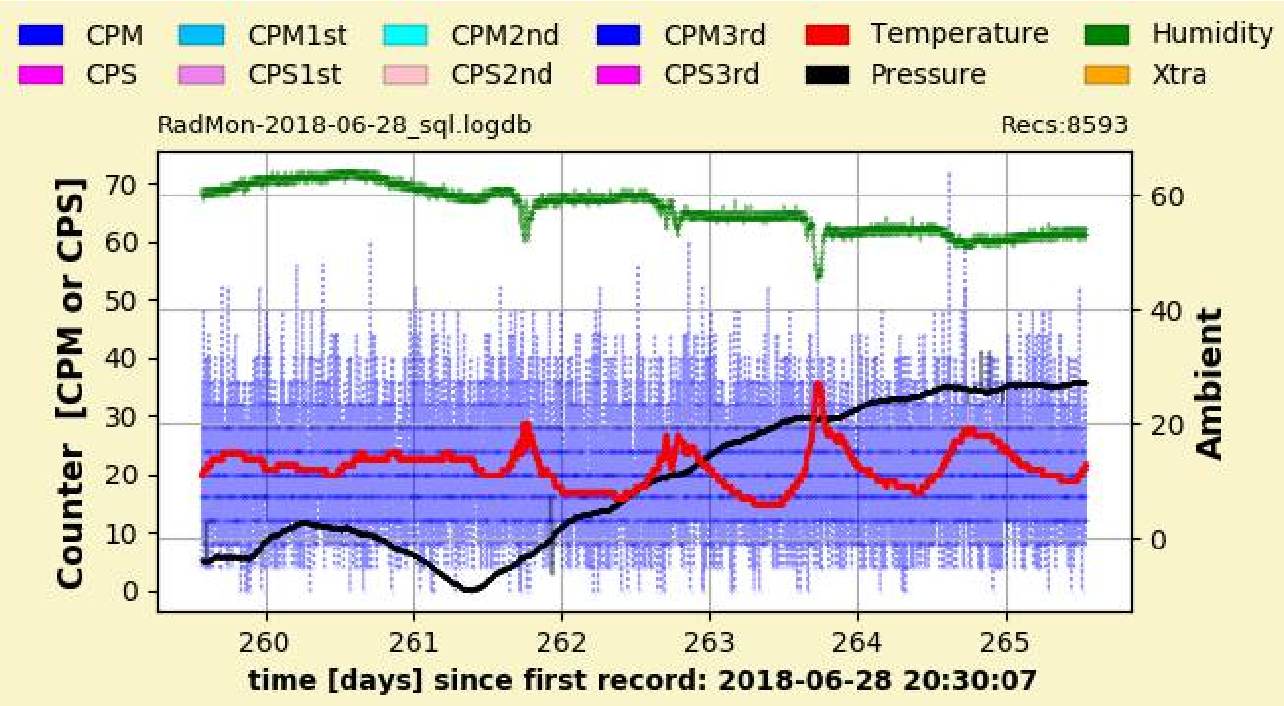


Fig 15 Long-Term Recording from an Outside RadMon+ Weather Station

LabJack Devices

Labjack is a company (https://labjack.com/) providing a range of data collection hardware, typically for laboratory and industrial purposes. Here the hardware device U3 in combination with the ei1050 probe for temperature and humidity is implemented.

Labjack provides a set of drivers and Python 3 based software to operate these devices.

The present Labjack installation in GeigerLog is only supported on Linux and perhaps Mac, but NOT on Windows!

For LabJack Python support see: https://labjack.com/support/software/examples/ud/labjackpython

INSTALLATIONS

To use the LabJack device you need the installation of (More details in the head of the Python code in file glabjack.py).
- the so called Exodriver
- the LabJackPython library
- the u3 Python module – included in the LabJackPython library package
- the ei1050 Python module – included in the LabJackPython library package

The Exodriver is here: https://labjack.com/support/software/installers/exodriver. The latest version is 2.6.0, compatible with Linux kernels from 2.6.28 onwards. Download, unzip and run:
`$ sudo ./install.sh`

LabJackPython is needed in the version 2.0.0 (Jan 2019) or later in order for compatibility with Python3! Available here: https://labjack.com/support/software/examples/ud/labjackpython

Unzip and install with: `$ sudo python3  setup.py  install`

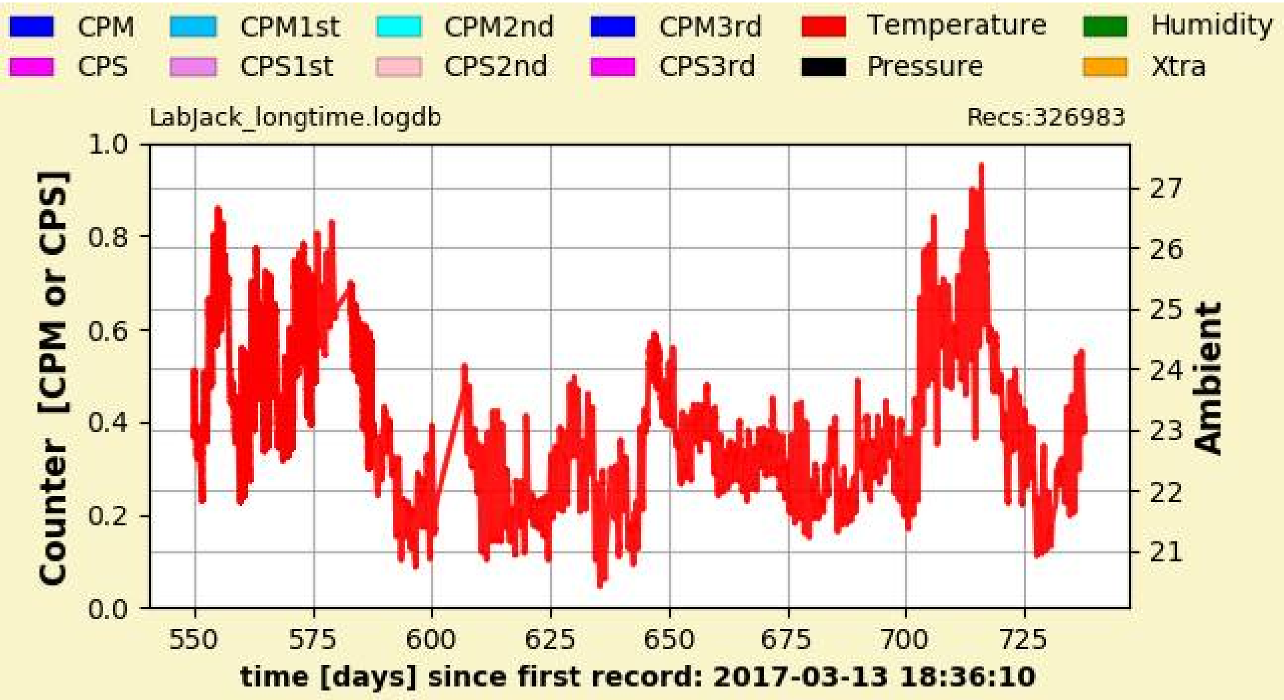


Fig. 16 Long-Term Recording -- 2 years -- with the LabJack device, showing Temperature only

Gamma-Scout Devices

The company Gamma-Scout (https://www.gamma-scout.com/en/) offers 4 different Geiger counters, of which the 3 devices ***Standard***, ***Alert***, and ***Rechargeable*** do NOT allow logging to a computer, and only allow a history download from (a rather small) internal memory to a computer.

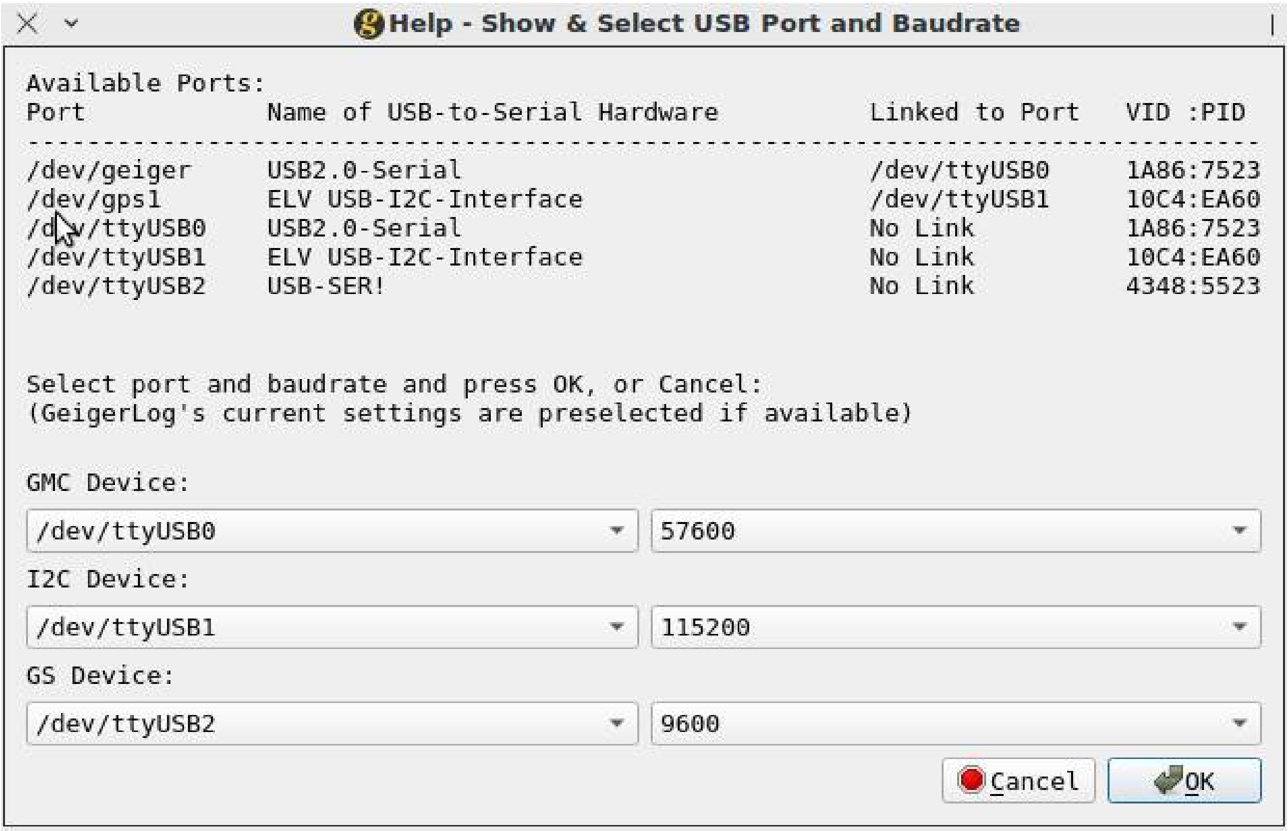
The 4th model ***Online*** allows some limited logging, but GeigerLog so far only supports a History download (https://www.gamma-scout.com/en/measures-radioactivity-easily-and-reliably/).

Installations

There is nothing to install. Just make sure that Gamma-Scout devices are activated in the configuration file `geigerlog.cfg` in the section **`GammaScoutDevice`**.

Connecting

Since GS devices use a USB-to-Serial driver, you might wish to also check the settings for the GS devices in the section **`GammaScoutSerialPort`**. However, if you are unfamiliar with the required settings, you can simply start GeigerLog and select menu **Help** → **Show & Select USB Port and Baudrate,** or use the blue toolbar button **Help Port**, to see this dialog box:



You can select your settings in this dialog and connect and run GeigerLog with your devices, but these settings are temporarily; they are maintained for this session only, and need to be reselected the next time you use GeigerLog.

It is more convenient to use the settings found and insert them into your configuration file!

History Download

From the menu choose **History → Gamma Scout Series → Get History from Device**, and the history will be downloaded and saved into a database.

If you have a *.dat file created with Gamma-Scout software (which contains a memory dump) you can use **History → Gamma Scout Series → Get History from Gamma-Scout Dat File** to load this file into a GeigerLog database, just as if were downloaded from a GS device.

Data Interpretation

The GS devices do not directly produce CPM and CPS data, but record the counts within intervals ranging from a minimum of 10 sec to a maximum of 1 week. Thus there is never a recording of true CPS, and true CPM only when the recording interval is 1 minute!

GeigerLog maps the true counts to variable CPM1st, and the interval to variable X. The variable CPM2nd holds the **calculated (!)** CPM. In this example the counter (index 9, 11) was set to 10 sec collection time, and only 2, or 1, resp, counts were seen, giving a calculated CPM of 12 and 6, resp.

In index lines 53 and 55 the counts were collected over 60 sec, so actual counts and calculated CPM were the same a 30 and 36, resp.

```
#>Index, DateTime,           CPM1st, CPM2nd,    X
      9, 2019-08-14 18:37:00,   2.0,   12.0, 10.0
     11, 2019-08-14 18:37:10,   1.0,    6.0, 10.0
      ~
     53, 2019-08-14 22:12:20,  30.0,   30.0, 60.0
     55, 2019-08-14 22:13:20,  36.0,   36.0, 60.0
```

Be aware that a Poisson test is possible ONLY on the TRUE counts! (Try it out!)

The GUI – Graphical User Interface

Menus

Menu items may be grayed out when currently not selectable. Some items have keyboard shortcuts in the form of CTRL-X; see the menus for the codes to be used in lieu of the X.

Menu – File

Commands to plot, print and save data, statistics, and exit the program

• Plot Log	Plot the log file data (if loaded)
• Plot History	Plot the history file data (if loaded)
• Plot Scatter	Plot one variable against another in a x-y-scatter plot
• Show Data as Shown in Plot	Show the Date&Time and variable values as currently shown in the plot
• Show Summary Statistics (SuSt)	Print summary statistics for all data currently shown in the plot to the NotePad
• Show Statistics	In a pop-up window show detailed statistics for the data currently shown and selected in the plot
• Show Poisson Test	In a pop-up window show a ‘Histogram with Poisson Fit’ for the data currently shown and selected in the plot
• Show FFT & Autocorrelation	In a pop-up window show a ‘FFT & Autocorrelation’ analysis of the data currently shown and selected in the plot
• Save NotePad to File	Save content of NotePad as text file `<current filename>.notes`
• Print NotePad	Print content of NotePad to a hardware printer or to a pdf file
• Exit	Exit the program (will be prevented if Logging is ongoing; stop Logging first)

Menu – Device

Commands related to the devices, their status,their configuration, and operating mode.

• Connect Devices	Connect computer with devices
• Disconnect Devices	Disconnect computer from devices
• Set Calibrations for Geiger Tubes	Set calibrations for all Geiger tubes temporarily

- Show Device Mappings — Show the mapping of variables to the activated devices

The submenus of the individual device series show up ONLY when these devices are activated in the configuration file `geigerlog.cfg`!

- Submenu: GMC-Series — Functions related to GMC-Devices
- Submenu: AudioCounter-Series — Functions related to AudioCounter Devices
- Submenu: I2CSensors-Series — Functions related to I2C-Sensor based devices
- Submenu: RadMon-Series — Functions related to RadMon Devices
- Submenu: AmbioMon-Series — Functions related to AmbioMon Devices
- Submenu: LabJack-Series — Functions related to LabJack based devices
- Submenu: Gamma-Scout-Series — Functions related to Gamma-Scout devices

Menu – Device – Submenu: GMC-Series

- Show Info — Prints some basic info about the device to the NotePad. This is the same info that is printed upon connection.
- Show Extended Info — Prints extended info about the device to the NotePad; see also Appendix D – The Device Configuration Meanings for some content included in info.
- Show Configuration Memory — Prints the device configuration as binary data in human readable format to the NotePad. Also see Appendix D – The GMC Device Configuration Meanings.
- Switch Power ON — Switches the device power ON (as if pressing the Power button on the device).
- Switch Power OFF — Switches the device power OFF (as if pressing the Power button on the device).
- Switch Alarm ON — Switches the device alarm ON.
- Switch Alarm OFF — Switches the device alarm OFF.
- Switch Speaker ON — Switches the device speaker ON.
- Switch Speaker OFF — Switches the device speaker OFF.
- Set History Saving Mode — Sets the device’s mode of History saving. Can be:
  - OFF (no history saving)
  - CPS, save every second
  - CPM, save every minute
  - CPM, save hourly average

	- CPS, save every second if exceeding threshold - CPM, save every minute if exceeding threshold
• Set Date + Time	Synchronizes computer and device time by setting the device's date and time to the computer time.
• Reboot	Reboots the device.
• FACTORYRESET	Does a factory reset. Your device customization is lost, and the internal memory is cleared.

Menu – Device – Submenu: AudioCounter-Series

- Show Info — Prints some basic info about the device to the NotePad. This is the same info that is printed upon connection.
- Show Extended Info — Additional info explaining how the audio pulses are interpreted
- Toggle Pulse Sound — Makes the audio clicks audible by toggling the AudioCounter Device sound output ON or OFF. Will be automatically switched off during logging
- Plot Pulse — Plot the audio pulse recording from the AudioCounter Device. Makes GeigerLog work like a digital-storage-oscilloscope showing the audio pulses. It provides option to analyze the pulses in the audio recordings

Menu – Device – Submenu: I2CSensors-Series

- Show Info — Prints some basic info about the device to the NotePad. This is the same info that is printed upon connection.
- Show Extended Info — Additional info on the device
- Reset System — Reset the I2C ELV dongle and sensors

Menu – Device – Submenu: RadMon-Series

- Show Info — Prints some basic info about the device to the NotePad. This is the same info that is printed upon connection.
- Show Extended Info — Additional info on the device

Menu – Device – Submenu: AmbioMon-Series

- Show Info — Prints some basic info about the device to the NotePad. This is the same info that is printed upon connection.
- Show Extended Info — Additional info on the device

- Configure — Configure settings of the AmbioMon device

Menu – Device – Submenu: LabJack-Series

- Show Info — Prints some basic info about the device to the NotePad. This is the same info that is printed upon connection.
- Show Extended Info — Additional info on the device

Menu – Device – Submenu: Gamma-Scout-Series

- Show Info — Prints some basic info about the device to the NotePad. This is the same info that is printed upon connection.
- Show Extended Info — Additional info on the device
- Set to Normal Mode — The device can be manually controlled at the device
- Set to PC Mode — The device can be controlled from the computer automatically set at connection, and required for downloads.

Menu – Log

Commands related to logging.

- Get Log or Create New One — Opens a dialog box where you can either select an existing file, or type in a new file name to create a new file. The file will be a database file with the extension *.logdb.

  If you select an existing file, new data will be **appended** to this file!

  After loading a file, it will always be plotted if it contains data, which can be plotted
- Get Log from CSV File — This allows you to load log-files which were created by older versions of GeigerLog (or by completely different programs, as long as the data are in a CSV format).

  Opens a dialog box where you can select an existing file with the extension *.log or *.csv. Then another dialog box opens, which allows you associate the data columns with the variables in the present GeigerLog version.

  A new database file will the be created.
- Set Log Timings — The log cycle in seconds can be set in a pop-up window. The cycle time must be at least 0.1s; shorter times cannot be entered

- Start Logging — Starts logging. Requires that
  1) a connection is made to the device,
  2) the device is powered on, and
  3) a log file is loaded

  The logged values will immediately be saved to the log file, printed to the LogPad, and plotted
- Stop Logging — Stops logging
- Quick Log — Start logging using the file default.log. The file will be emptied, before logging starts. If you want to continue logging into a previously selected default.log file, then choose Start Logging instead
- Add Comment to Log — Adds a comment to the log file; does not disturb logging or graphing.
- Show Log Data — Prints the log data to the NotePad.
- Show Log Data Excerpt — Prints only the first and last few lines of the log, helpful for quick inspection
- Show Log Tags/Comments — Print only records from current log containing tags or comments to the NotePad
- Save Log Data into *.log file (CSV) — Save all records from current log into a CSV file with extension 'log'

Menu – History

Commands related to downloading the history stored on the internal memory of a Geiger counters. Applicable only to devices, which support this feature, currently GMC and GS devices.

- Get History from Database — Opens a dialog box and lets you select an existing database file, loads it and plots the data
- Get History from CSV File — Lets you select an existing CSV file – created e.g. by an earlier version of GeigerLog as a '.his' file – and saves it into a database, and then plots the data.
- Add Comment to History — Adds a comment to the history database
- Submenu: GMC-Series — History functions related to GMC Devices
- Submenu: Gamma-Scout Series — History functions related to Gamma-Scout Devices
- Show History Data — Print history data as parsed from binary data to the NotePad
- Show History Data Excerpt — Prints the first and last few lines of the parsed data to the NotePad

- Show History Tags/Comments — Prints only those lines from the parsed data to the NotePad, which contain tags or comments. These are mostly Date&Time stamps, but also ASCII tags, which are comments entered directly at the GMC Geiger counter via its Main Menu → Save Data → Note/Location.
- Show History Data with Parse Comments — Show History Data including extended Parse Comments as created by GeigerLog
- Save History Data into *.his file (CSV) — Save all records from current history into a CSV file with extension 'his'

Menu – History – Submenu: GMC Series

- Get History from Device — Opens a dialog box where you can select either an existing file, or type in a new name to create a new file. If you select an existing file, this file will be overwritten and its present content will be lost!

  This file is a database file with the extension *.hisdb.

  GeigerLog reads the data from the internal memory of the Geiger counter, and stores an exact binary copy in the database.

  GeigerLog then parses the binary data and creates a log of the count rates. These data will be plotted.
- Get History from GMC Binary File — Lets you select an existing binary file – created e.g. by an earlier version of GeigerLog or by a different program – and saves it into a database, parsed the file to create a log of the count rates, and plots the data
- Show History Binary Data Bytecount — Show counts of bytes in history binary data
- Show History Binary Data — Print history binary data in human readable form to the NotePad
- Show History Binary Data Excerpt — Prints the first and last few lines of history binary data in human readable form to the NotePad
- Show History Binary Data as FF Map — Show History Binary Data as a map highlighting the locations of bytes with FF value

- Save History Binary Data to File — Save the history binary data as a *.bin file, i.e. the binary data are extracted from the database and saved into a binary '.bin' file, compatible with earlier versions of GeigerLog.

Menu – History – Submenu: Gamma-Scout Series

- Get History from Device — Opens a dialog box where you can select either an existing file, or type in a new name to create a new file. If you select an existing file, this file will be overwritten and its present content will be lost!

  This file is a database file with the extension *.hisdb.

  GeigerLog reads the data from the internal memory of the Geiger counter, and stores an exact binary copy in the database.

  GeigerLog then parses the data and creates a log of the count rates. These data will be plotted.
- Get History from GS Dat File — Lets you select an existing *.dat file created by Gamma-Scout software as a download of the counter's memory, loads it, and saves it into a database, parses the file to create a log of the count rates, and plots the data.
- Show History Dat Data — Print history Dat data, as they were downloaded from the counter's memory, to the NotePad.
- Save History Data to Dat File — Save the history data as a Gamma-Scout '.dat' file, i.e. the data as downloaded from the counter are saved in the Dat file format.

Menu – Web

- Update Radiation World Maps — Upload your current data to the Radiation World Maps, see Radiation World Maps on page 55.

Menu – Help

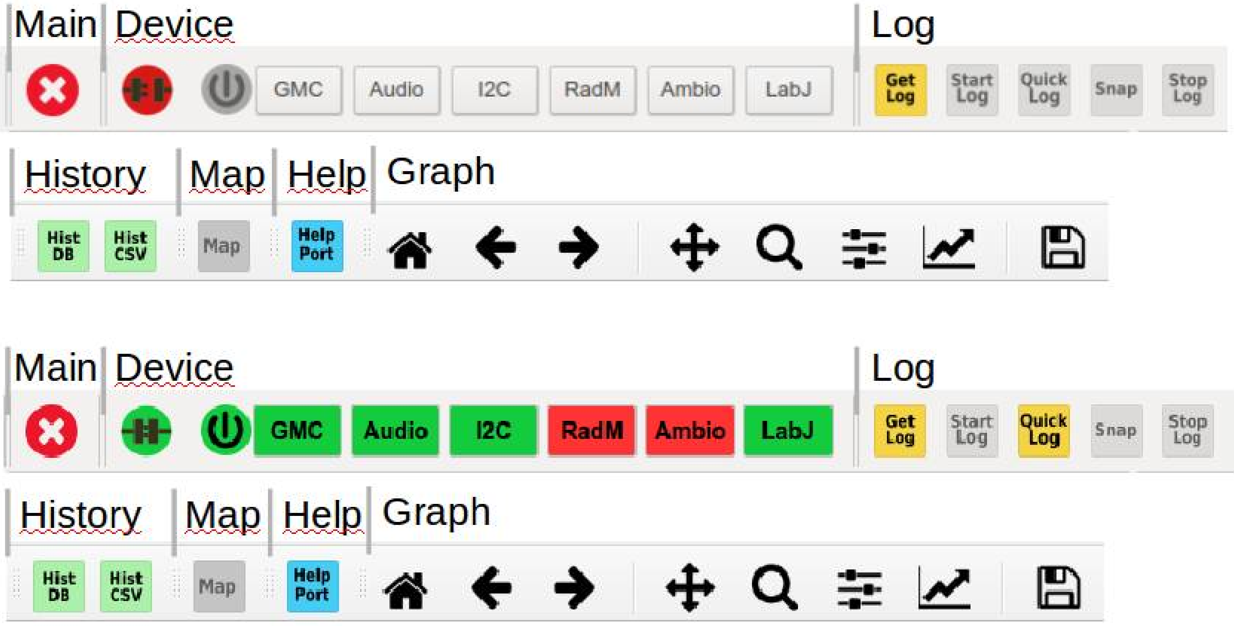
Some helpful information for running GeigerLog.

- Quickstart — A very short GeigerLog Manual
- GeigerLog Manual — Opens the GeigerLog Manual. Will attempt to open it locally, but if not available then does it online

• Devices' Firmware Bugs	Some info on firmware bugs perhaps of relevance to the user, and workarounds
• Radiation World Maps	A brief introduction into the use of the Radiation World Maps
• Occupational Radiation Limits	Info on occupational radiation limits of USA and Germany, and links for extended info.
• View Current Scaling	View the current value- and graph-scaling settings
• Show & Select USB Port and Baudrate	Pops up a dialogue box with details on existing USB ports and hardware, and allows you to select port and baudrate for a GMC counter, an I2C device, and a GS device.
• Autodiscover GMC-Connected USB Port	GeigerLog makes an attempt to determine your port and baud rate for a GMC Geiger counter automatically, and shows the report in a pop-up dialog box. If a single GMC Geiger Counter was found, you can take the found settings for this session and are given advice how to make this permanent. Otherwise info is given on how to proceed. See Appendix B – Connecting GMC Device and Computer on page 59 for more details.
• About GeigerLog	A brief introduction to GeigerLog, as well as version and legal information

Toolbars

The six individual toolbars **Main**, **Device**, **Log**, **History, Map,** and **Graph** are combined into a single toolbar, see fig. 2. If preferred, they can be separated and relocated on the screen by grabbing their vertical bars on the left and moving them.



The toolbars

Top : non-connected status *Bottom: connected status*

The status in this picture is that all possible devices are activated (in the configuration file), and all shown in green color are connected, while for those in red color the connection attempt has failed.

- Main - Exit GeigerLog
- Device - Symbol 'plug': Toggle connection of all activated devices
  - Device buttons are shown here **only if** the respective device is activated in the configuration file `geigerlog.cfg`.
  - Symbol 'on/off button': belongs only to the GMC device. Green color indicates that a GMC Geiger counter is powered on (red = power-off)
  - Rectangular buttons indicate activated devices. Color = green: successully connected, otherwise color= red. The figure shows:
  - All possible devices are activated (done in the configuration file)
  - A GMC Geiger counter is connected (green color)
  - An AudioCounter device is connected (green color)
  - An I2CSensors device is connected (green color)
  - A RadMon device is activated, but its connection failed (red color)
  - An AmbioMon device is activated, but its connection failed (red color)
  - A Labjack device is connected (green color)

- Log - Get a Log File
  - Start Logging
  - Start a Quick Log
  - Snap a single record during logging
  - Stop Logging

- History - Get History from database
  - Get History from CSV file

- Map - Upload your current data to the Radiation World Maps, see Radiation World Maps on page 55.
  You must be logging in order for the Map icon to become active (=blue)

- Help - Open the **Help – Show & Select USB Port and Baudrate** dialog to select the settings for the GMC, I2C, and GS devices (if any are present)

- Graph - Reset original view
  - Back to previous view
  - Forward to next view
  - Pan axes with left mouse, zoom with right
  - Zoom to rectangle
  - Configure subplots
  - Save the figure
  - Edit curves, line, and axes parameters

Miscellaneous

Starting GeigerLog with Options

You can start GeigerLog with options, typically used for temporary adjustments. Otherwise you might prefer to customize the configuration file `geigerlog.cfg`. To see the available options, start GeigerLog with ‘`geigerlog -h`’. You will get this printed out to the terminal:

```
Usage:  geigerlog [Options] [Commands]

By default, data files will be read-from/written-to the
data directory "data", a subdirectory to the program directory

Options:
    -h, --help          Show this help and exit.
    -d, --debug         Run with printing debug info.
                        Default is debug = False.
    -v, --verbose       Be more verbose.
                        Default is verbose = False.
    -w, --werbose       Be even more verbose.
                        Default is werbose = False.
    -V, --Version       Show version status and exit.
    -P, --Portlist      Show available USB-to-Serial ports
                        and exit.
    -R, --Redirect      Redirect stdout and stderr to
                        file geigerlog.stdlog (for debugging).
    -s  --style name    Sets the style; see also manual and
                        Command: 'showstyles'.
                        Default is set by your system.

Commands:
    showstyles          Show a list of styles avail-
                        able on your system and exit.
                        For usage details see manual.
    keepFF              Keeps all hexadecimal FF
                        (Decimal 255) values as a
                        real value and not an 'Empty'
                        one. See manual in chapter
                        on parsing strategy.
    devel               Development settings; careful!
                        see program code.

To watch debug and verbose output start the program from the
command line in a terminal. The output will print to the terminal.
With the Redirect option all output - including Python error
messages - will go into the redirect file geigerlog.stdlog.
```

Of interest for debugging is the option ‘`-R`’ (or ‘`--Redirect`’). While the program log file `geigerlog.proglog` has all output from GeigerLog, it does not have any error messages from Python itself, which are essential for debugging. With the redirect option another log file `geigerlog.stdlog` is created, which contains these as well. However, there won’t be any live output to the terminal at all, which makes this option inconvenient for normal use.

Radiation World Maps

Several web sites exist, which attempt to show a worldwide map of the **BACKGROUND** radioactivity, hoping to be of help to the people in case of a nuclear emergency, which will result in elevated levels of radioactivity. Some are run by governments, others by enthusiastic hobbyists.

Among the latter ones are:

- gmcmap.com - This is the one supported by GQ Electronics
- radmon.org - Presently down after being hacked
- safecast.org - Accepting radiation as well as air quality data

Currently only GQ's GMCmap is supported by GeigerLog; others may follow.

GQ suggests to use your Geiger counter (versions with WiFi, i.e. GMC-320+V5, GMC-500, GMC-600 series) to directly update their website. This is actually not such a good idea, see below.

But you can also support their world map using GeigerLog, and not only provide more meaningful data, but use any of their non-WiFi counters – old and new ones – just as well. If you want to contibute to gmcmap.com, you need to register there. This provides you with a UserID and a CounterID. Enter both into the respective fields in the GeigerLog configuration file '`geigerlog.cfg`' under the heading 'Worldmaps'. That's it!

When you are logging, the toolbar icon 'Map' turns blue , aka it becomes enabled (as well as the menu entry Web → Update Radiation World Map), so only fresh data can go into the world map. Click the icon and you'll be presented with a dialogue box, showing you the data you would be uploading if you clicked ok. But you could also click cancel. Obviously, for this upload to succeed you need to have an active internet connection at your computer!

You will see a confirmation printed to the NotePad, including the response of the website.

A word of caution

There are several problems with at least the GQ world map and the way data is sent to them.

CPM: The property depicted on the map is CPM, which is the worst possible base on which to compare different counters, which may have different tubes and even different tube numbers, and therefore totally different calibration factors to translate from CPM to a true dose rate like measured in µSv/h. This is like a worldwide reporting of temperatures as either Fahrenheit, or Celsius, or Reaumur but not telling which is which. The only meaningful basis for comparisons is the dose rate based on units of Sievert per time interval (µSv/h, or nSv/h).

Quality Control: As far as I can see there is no quality control of the data! Nothing prevents users from putting a strong radioactive source in front of their detector, and pushing these data to the web. In fact, you don't even need a counter, and don't even need GeigerLog, but can enter any data you wish manually! I don’t want to mess with GQ’s map, so I haven’t tried to enter things like CPM=9999. But if you did something like that inadvertently you would discover that there does not seem to be a way to retract any such wrongly sent data.

Poor data will quickly destroy any value of those sites.

Instantaneous CPM: It is a bit more subtle, but diminishes the data quality nevertheless. GQ's potential upload is: CPM, Average-CPM, µSv/h reading. The latter two are optional. Which lets me to conclude that the attended CPM upload is the instantaneous CPM of the counter.

Unfortunately, Geiger counter readings fluctuate quite significantly. Thus when individual, single readings are posted, the values may be significantly higher or lower than the average, suggesting changes that don't exist. The fluctuation is largest at low count rates [24]), hence the reports of background rates are the most impacted: for a CPM=20 average background, 5% of the values can be expected to be greater than CPM=28 or smaller than CPM=10. That is almost a 3fold difference!

GeigerLog will always send averages-only as CPM values, but allows the user to determine the number of data points, which are used for the average. It does this be averaging ALL data you see in the plot in the moment you press the Map button. Thus you can use the mouse buttons to easily select an appropriate stretch of data; in the extreme, this stretch could be a single data point! And GeigerLog uses this so determined average for both CPM and ACPM, and calculates µSv/h based on it. I suggest to have values collected for at least 30 minutes, more is better, before sending anything to the maps.

Governmental sites like this Swiss site [25]) provide only DAILY averages of quality controlled data!

Occupational Radiation Limits

The exposure to radiation is strongly regulated all over the world. With respect to the Radiation World Maps it is quite interesting to compare regulations in different countries. As examples, the occupational limits are given for USA and Germany.

"Occupational" refers to people working in fields with typically higher exposure to radiation compared to the average person, like medical people applying X-rays, workers in nuclear power plants, people in aviation, people in mining.

Of the many limits specified, only the yearly and lifelong exposures are given here; the links will guide you to sites with more extensive specifications.

	Germany	**USA**
Yearly exposure	20 mSv	50 mSv
Lifelong exposure	400 mSv	2350 mSv
Links	BfS Grenzwerte	OSHA

The differences are quite significant; see details in the links.

24 it decreases with 1/SQRT(count rate), see Poisson Distribution in GeigerLog - Potty Training for Your Geiger Counter

25 https://www.naz.ch/de/aktuell/tagesmittelwerte.shtml

Problems and Bugs

If your attempts to start GeigerLog fail, perhaps because the distribution you are using has different defaults, start GeigerLog from the terminal/command line, and look for error messages. Look through these error messages to find out if e.g. any modules are missing and what these modules are. Look through Appendix H – Installation on page 75 for more guidance.

If you do encounter any bugs or problems please report via the project GeigerLog site at Source-Forge: https://SourceForge.net/projects/geigerlog/. I will need the file `geigerlog.stdlog`, which will be created when GeigerLog is started with the '`-R`' option, see Error: Reference source not found on page Error: Reference source not found.

On SourceForge you also have the option to send me an email.

References

Geiger-Müller tubes - Introduction, Centronic ISS.1 (further details unknown).
Downloaded April 2017 from:
https://SourceForge.net/projects/gqgmc/files/gqgmc/Geiger_Tube_theory.pdf/download
also available here: http://qa.ff.up.pt/radioquimica/Bibliografia/Diversos/geiger_tube_theory.pdf

Accurate Determination of the Deadtime and Recovery Characteristics of Geiger-Muller Counters, Louis Costrel, U.S. Department of Commerce, National Bureau of Standards, Research Paper RP1965, Volume 42, March 1949, Part of the Journal of Research of the National Bureau of Standards, http://nvlpubs.nist.gov/nistpubs/jres/42/jresv42n3p241_A1b.pdf

GQGMC, Documentation by Phil Gillaspy
https://sourceforge.net/projects/gqgmc

GQ-RFC1201, GQ Geiger Counter Communication Protocol, Ver 1.40, Jan-2015,
by GQ Electronics LLC, https://www.gqelectronicsllc.com/download/GQ-RFC1201.txt

GeigerLog - Potty Training for Your Geiger Counter, by ullix,
https://SourceForge.net/projects/geigerlog/

GeigerLog - Going Banana, by ullix,
https://SourceForge.net/projects/geigerlog/

Appendix A – Look & Feel

GeigerLog uses some resources which exist on your computer independently from GeigerLog. This is mainly the “style”, but also the “fonts” available on a system. Both largely determine the Look & Feel of a software.

They may differ between computers.

Style

Generally the default style will be ok. But if it doesn’t please you, select a different one. To get a list of styles available on your computer, start GeigerLog with:

```
./geigerlog showstyles
```

Your output should be similar to this [26]):

```
Linux:                Windows, Fusion
Windows:              WindowsVista, Windows, WindowsXP, Fusion
```

To use a style, let’s say `'Windows'`, start GeigerLog like this:

```
./geigerlog -s Windows
```

Fonts

GeigerLog will select suitable fonts; they cannot be selected by the user.

26 The previously possible styles Breeze, Cleanlooks, Plastique, Windows, GTK+, and other, are no longer available under the modern PyQt5 toolkit in default installations

Appendix B – Connecting GMC Device and Computer

BACKGROUND: Device and computer are connected with a USB cable, but the connection is actually a classic serial connection. The translation between USB and serial is done by an USB-to-Serial chip in the electronics of the counter. And while serial connections today are much faster than in the good old days of the teletype, they are slow by today's standards. In the GMC series the serial speed is in the order of 0.1 MBit/s, while USB2 is nominal 480 Mbit/s and USB3 even in the Gigabit/s range. With respect to logging, the speed is sufficient, but for other actions a faster speed would be welcome.

For a successful connection you need to know the name the computer has given to the serial port, the baud rate of the device, and have a driver installed.

The serial port, the baud rate and the driver

The baud rate is set at the GMC device itself. To look it up at the counter go to its Main Menu → Others → Comport Baud Rate. The default is 57600 (older devices) or 115200 (newer devices). I suggest to keep the default setting (I experienced occasional read errors with a GMC-300 device, which seemed to have to do with the baud rate; and sometimes the counter chokes when things go too fast).

On Linux the driver is already part of the system. On Windows and Mac a driver must be installed. Drivers may be available for download on the GQ website [27]).

Depending on circumstances, a different hurdle may exist for Linux, as a regular users (non-administrator) may not have the read- and write-permissions to work with the serial port. See Appendix C – HOWTO deal with read and write permissions for the serial port when on Linux for a HOWTO on dealing with read and write permissions for the Serial Port on Linux.

I cannot give advice for a Windows or Mac system. However, the GeigerLog program itself may be able to help all users finding the right configuration.

Using GeigerLog to find the Serial Port Settings

Connect the Geiger counter with the computer. Start GeigerLog and click menu **Help → Autodiscover connected USB Port**. GeigerLog will test all available ports with all baud rates and report result into a pop-up dialog box. On a Linux system the result may be as shown in fig. 9.

> **WARNING:** If you have other devices – besides any Geiger counter – connected with a **USB-to-Serial** adapter (which is what your counter has built-in) it is possible that the other devices are significantly disturbed by the USB Autodiscovery!
>
> It is recommended to halt or switch off those other devices BUT LEAVE THEIR USB CABLE CONNECTED, and only then start the USB Autodiscovery.

27 http://www.gqelectronicsllc.com/comersus/store/download.asp

Any other native USB devices (mouse, keyboard, printer, ...) won't be impacted by the test.

This tells you that the Geiger counter was found at Port: /dev/ttyUSB0 with Baudrate: 57600, while the current setting of GeigerLog (bottom) is: Port: /dev/ttyUSB1 and Baudrate: 115200 .

Fig. B1 USB Autodiscovery Pop-up Box

You could now click the OK button, and the proper setting becomes active in this session, and an attempt will be made to connect to the Geiger Counter. But after a restart, you would have to repeat the procedure.

On the next start you could correct this by starting GeigerLog with these options:

```
./geigerlog -p /dev/ttyUSB0 -b 57600
```

For a permanent correction edit the section **Serial Port** of the configuration file 'geigerlog.cfg' and save, and then start GeigerLog without options (modified lines in red) :

```
[Serial Port]
# NOTE: settings in this section will be overwritten by command line options
# default = /dev/ttyUSB0
port      = /dev/ttyUSB0

# baudrates 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200
# the device must have been set to the selected baudrate!
# default = 115200
baudrate  = 57600
```

On a Windows system it is similar, but instead of '/dev/ttyUSBX', with X = 0, 1, 2, … it will say 'COMX', with X = 0, 1, 2, … so starting with a temporary correction might be:

```
geigerlog -p COM3 -b 57600
```

and the configuration file modified to:

```
port      = COM3
baudrate  = 57600
```

When running GeigerLog you can get a brief info on USB Port, Baud rate, Logging, History and Graphic by clicking menu **Help → Quickstart**.

Appendix C – HOWTO deal with read and write permissions for the serial port when on Linux

After you have connected the Geiger counter to the USB port, open a terminal and run this command:

```
ls -al /dev/ttyUSB*
```

the output is like :

```
crw-rw---- 1 root dialout 188, 0 Feb 26 12:16 /dev/ttyUSB0
```

It shows that the Geiger counter is connected to port '/dev/ttyUSB0' and that only the user root and all users in group dialout have read and write permissions (rw). Everybody else can neither read nor write!

Unless you are logged in as root (which you shouldn't be doing for normal work) you can only use the device if you belong to the group dialout. To see whether you do, enter in a terminal (assuming your username is 'myname'):

```
groups myname
```

giving an output listing of all groups you are a member of, like:

```
myname : myname cdrom sudo dip plugdev lpadmin
```

There is no group dialout listed, and hence you have no permission for the serial port and cannot work with the Geiger counter.

You have 3 options to overcome this problem, of which the 3rd is the recommended one:

1) Change permissions

In a terminal run `'sudo chmod 666 /dev/ttyUSB0'.` Follow by `'ls -al /dev/ttyUSB0'`and you see:

```
crw-rw-rw- 1 root dialout 188, 0 Feb 26 12:34 /dev/ttyUSB0
```

Now everyone has read and write permission. Security concerns may not be relevant here, but the problem is that you have to do this every time you unplug/replug the device!

2) Make yourself a member of group 'dialout'

To become a member of the dialout group, enter in a terminal:

```
sudo usermod -a -G dialout myname
```

You will need to logout and log back in to see your new group added:

```
groups myname
```

results in:

```
myname : myname dialout cdrom sudo dip plugdev lpadmin
```

This change is permanent; also survives a reboot.

But what if '`ls -al /dev/ttyUSB*`' gets you:

```
crw-rw---- 1 root dialout 188, 0 Feb 26 12:58 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Feb 26 12:59 /dev/ttyUSB1
```

This tells you that now two USB-to-Serial devices are connected to your computer. Obviously you can't tell from this listing which one is the new and which the old one. You'll have to try it out. With even more USB-to-Serial devices connected, it becomes even more complicated. And after a reboot, the order of the devices may have changed!

3) Take advantage of udev rules

In a terminal issue (as regular user):

```
lsusb
```

to get something similar to:

```
   Bus 002 Device 002: ID 8087:8000 Intel Corp.
   Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
   Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
*  Bus 003 Device 004: ID 4348:5523 WinChipHead USB->RS 232 adapter with Prolifec PL 2303 chipset
   Bus 003 Device 003: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
*  Bus 003 Device 002: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter
```

The listing shows all USB devices of the computer, of which some belong to its inner circuitry. The two USB-to-Serial adapter are marked with an asterisk on the left; the latter one is from the Geiger counter. Its ID is 1a86:7523, the first 4 hex digits being the vendor ID, the other 4 the product ID.

A udev rule allows the computer to recognize the connection of a device by this ID, and make certain settings and configurations, like giving read and write permissions.

Create a file containing nothing but these two lines:

```
# Comment: udev rule for GQ Electronics's GMC-300 Geiger counter
SUBSYSTEM=="tty", KERNEL=="ttyUSB*", ATTRS{idVendor}=="1a86", MODE:="666", SYMLINK+="geiger"
```

and save (you must be root to do this) as file '55-geiger.rules' in directory '/etc/udev/rules.d'. Then restart your computer (or issue the command '`sudo udevadm control --reload-rules`'). Then unplug and replug your Geiger counter device. You will now always find your device at port '/dev/geiger', irrespective of how many other devices are connected, and to which /dev/ttyUSB* !

HOWEVER: The USB-ID belongs to the USB-to-serial converter chip installed in the Geiger counter. And since (to my knowledge) GQ is using the same chip an all Geiger versions, this simple rule will not allow to distinguish between them! You'll probably have to resort to option 2 above, and figure out, which /dev/ttyUSBX with X=1, 2, 3, ... belongs to which device! Not to mention that likely a million other devices may also be using the very same chip ...

This was tested on Ubuntu Mate 16.04.02 with kernel 4.8.0-39-generic.

Appendix D – The GMC Device Configuration Meanings

The device configuration of the GMC-300 series is read-out as 256 bytes of binary information. Its meaning is reported here: http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4447 . However, this list is not consistent with observed values at device 'GMC-300E Plus' with firmware 'GMC-300Re 4.20'. See here for even more differences: https://SourceForge.net/projects/gqgmc/files/gqgmc/GQ-GMC-ICD.odt/download

The GMC-500 and GMC-600 series of Geiger counters have a different configuration which in addition is twice as long at 512 bytes.

GQ has recently disclosed the configuration of the 500 and 600, but some details remain unclear so far. (see: http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4948), and at least 2 firmware bugs on the 500 series were discovered (see discussion in the topic). Please, report any problems via SourceForge (see Problems and Bugs on page 57).

The following list applies only to the 300 series:

```
CFG data Offset table. Starts from 0
Values in BOLD are read and/or set in GeigerLog
===============================================
PowerOnOff, //to check if the power is turned on/off intended
AlarmOnOff, //1
SpeakerOnOff,
GraphicModeOnOff,
BackLightTimeoutSeconds,
IdleTitleDisplayMode,
AlarmCPMValueHiByte, //6
AlarmCPMValueLoByte,
CalibrationCPMHiByte_0,
CalibrationCPMLoByte_0,
CalibrationuSvUcByte3_0,
CalibrationuSvUcByte2_0, //11
CalibrationuSvUcByte1_0,
CalibrationuSvUcByte0_0,
CalibrationCPMHiByte_1,
CalibrationCPMLoByte_1, //15
CalibrationuSvUcByte3_1,
CalibrationuSvUcByte2_1,
CalibrationuSvUcByte1_1,
CalibrationuSvUcByte0_1,
CalibrationCPMHiByte_2, //20
CalibrationCPMLoByte_2,
CalibrationuSvUcByte3_2,
CalibrationuSvUcByte2_2,
CalibrationuSvUcByte1_2,
CalibrationuSvUcByte0_2, //25
IdleDisplayMode,
AlarmValueuSvByte3,
AlarmValueuSvByte2,
AlarmValueuSvByte1,
AlarmValueuSvByte0, //30
AlarmType,
SaveDataType,
SwivelDisplay,
```

```
    ZoomByte3,
    ZoomByte2, //35
    ZoomByte1,
    ZoomByte0,
    SPI_DataSaveAddress2,
    SPI_DataSaveAddress1,
    SPI_DataSaveAddress0, //40
    SPI_DataReadAddress2,
    SPI_DataReadAddress1,
    SPI_DataReadAddress0,
    PowerSavingMode,
    Reserved, //45
    Reserved,
    Reserved,
    DisplayContrast,
    MAX_CPM_HIBYTE,
    MAX_CPM_LOBYTE, //50
    Reserved,
    LargeFontMode,
    LCDBackLightLevel,
    ReverseDisplayMode,
    MotionDetect, //55
    bBatteryType,
    BaudRate,
    Reserved,
    GraphicDrawingMode,
    LEDOnOff,
    Reserved,
    SaveThresholdValueuSv_m_nCPM_HIBYTE,
    SaveThresholdValueuSv_m_nCPM_LOBYTE,
    SaveThresholdMode,
    SaveThresholdValue3,
    SaveThresholdValue2,
    SaveThresholdValue1,
    SaveThresholdValue0,
    Save_DateTimeStamp, //this one uses 6 byte space
```

The following list applies only to the 500 and 600 series: (from: http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4948)

The GMC-500 and GMC-600 still accept configuration commands same as GMC-320, no change. But GMC-500 and GMC-600 extended (added new) commands for new features.

Here is the latest configuration data structure in C code on GMC-500 and GMC-600:

typedef enum {
CFG_PowerOnOff,
CFG_AlarmOnOff, //1
CFG_SpeakerOnOff,
CFG_IdleDisplayMode,
CFG_BackLightTimeoutSeconds,
CFG_IdleTitleDisplayMode,
CFG_AlarmCPMValueHiByte, //6
CFG_AlarmCPMValueLoByte,
CFG_CalibrationCPMHiByte_0,
CFG_CalibrationCPMLoByte_0,
CFG_CalibrationuSvUcByte3_0,
CFG_CalibrationuSvUcByte2_0, //11
CFG_CalibrationuSvUcByte1_0,

CFG_CalibrationuSvUcByte0_0,
CFG_CalibrationCPMHiByte_1,
CFG_CalibrationCPMLoByte_1, //15
CFG_CalibrationuSvUcByte3_1,
CFG_CalibrationuSvUcByte2_1,
CFG_CalibrationuSvUcByte1_1,
CFG_CalibrationuSvUcByte0_1,
CFG_CalibrationCPMHiByte_2, //20
CFG_CalibrationCPMLoByte_2,
CFG_CalibrationuSvUcByte3_2,
CFG_CalibrationuSvUcByte2_2,
CFG_CalibrationuSvUcByte1_2,
CFG_CalibrationuSvUcByte0_2, //25
CFG_IdleTextState,
CFG_AlarmValueuSvByte3,
CFG_AlarmValueuSvByte2,
CFG_AlarmValueuSvByte1,
CFG_AlarmValueuSvByte0, //30
CFG_AlarmType,
CFG_SaveDataType,
CFG_SwivelDisplay,
CFG_ZoomByte3,
CFG_ZoomByte2, //35
CFG_ZoomByte1,
CFG_ZoomByte0,
CFG_SPI_DataSaveAddress2,
CFG_SPI_DataSaveAddress1,
CFG_SPI_DataSaveAddress0, //40
CFG_SPI_DataReadAddress2,
CFG_SPI_DataReadAddress1,
CFG_SPI_DataReadAddress0,
CFG_nPowerSavingMode,
Reserved_1, //45
Reserved_2,
Reserved_3,
CFG_nDisplayContrast,
CFG_MAX_CPM_HIBYTE,
CFG_MAX_CPM_LOBYTE, //50
Reserved_4,
CFG_nLargeFontMode,
CFG_nLCDBackLightLevel,
CFG_nReverseDisplayMode,
CFG_nMotionDetect, //55
CFG_bBatteryType,
CFG_nBaudRate,
Reserved_5,
CFG_nGraphicDrawingMode,
CFG_nLEDOnOff, //60
Reserved_6,
CFG_nSaveThresholdValueuSv_m_nCPM_HIBYTE,
CFG_nSaveThresholdValueuSv_m_nCPM_LOBYTE,
CFG_nSaveThresholdMode,
CFG_nSaveThresholdValue3, //65
CFG_nSaveThresholdValue2,
CFG_nSaveThresholdValue1,
CFG_nSaveThresholdValue0,

CFG_SSID_0,
//...
CFG_SSID_31 = CFG_SSID_0 + 31, //68 + 31

CFG_Password_0, //100
//...
CFG_Password_31 = CFG_Password_0 + 31, //100 + 31

CFG_Website_0, //132
//....
CFG_Website_31 = CFG_Website_0 + 31, //132 + 31

CFG_URL_0, //163
//....
CFG_URL_31 = CFG_URL_0 + 31, //163 + 31

CFG_UserID_0, //195
//..........
CFG_UserID_31 = CFG_UserID_0 + 31, //195+31

CFG_CounterID_0, //227
//....
CFG_CounterID_31 = CFG_CounterID_0 + 31, //227 + 31

CFG_Period, //259
CFG_WIFIONOFF, //260
CFG_TEXT_STATUS_MODE,

/
CFG_Save_DateTimeStamp, //this one uses 6 byte space

CFG_MaximumCFGBytes,

}EEPROMDATAT;

ZLM: For GMC-500, GMC-600 history data C code structure:
(this should be same as GMC-300, no change)
In history data, it start with 0x55AA00 prefixed for timestamp and followed by the date time data. and then always followed by 0x55AA and one of the bellow data length byte.

typedef enum
{
YYMMDDHHMMSS, // Time Stamp
DOUBLEBYTE_DATA, //the data are double bytes
THREEBYTE_DATA, //the data are three bytes
FOURBYTE_DATA, //the data are four bytes
LOCATION_DATA, //the data is a text string,the first byte data is the length of the text, followed by the text

TOTAL_EEPROM_SAVE_TYPE

}HistoryDataFormatMarkingT;

Also, the 0x55AA also can follow a one of following history data type:
typedef enum
{

```
SAVEOFF,

SECONDLY, //must be save value with TOTAL_EEPROM_SAVE_TYPE
MINUTETLY, //must be save value with TOTAL_EEPROM_SAVE_TYPE
HOURLY, //must be save value with TOTAL_EEPROM_SAVE_TYPE
SaveByThresholdSecond, //only save the data if exceed the preset threshold value
SaveByThresholdMinute, //only save the data if exceed the preset threshold value

TotalSavedType

}SaveDataTypeT;
```

Appendix E – GMC Device: Internal Memory, Storage Format and Parsing Strategy

There is no official document from GQ on the storage format, but it is well described by user Phil Gillaspy in this document https://SourceForge.net/projects/gqgmc/files/gqgmc/GQ-GMC-ICD.odt/download. Other info comes from the analysis of the memory content using this GeigerLog program and a GMC-300E+ device.

The internal memory of the Geiger counters is handled like a ring-buffer. The device begins to write at the bottom, and fills the memory up. Once it reaches the top, it continues at the bottom and fills up again, overwriting the previous history. This principle in combination with the storage format creates some headaches for parsing, i.e. the method through which a log file can be created from reading and interpreting the data.

Let's start with the memory being completely erased - like after a factory reset, or a manual 'Erase Saved Data' command at the counter itself. Every single byte of the memory is set to the 'empty' value, which is hexadecimal FF, decimal 255. One problem already: you can also have a measured value of 255 and cannot distinguish between the two!

Date & Time Stamp

Once the memory is erased, the very first thing the counter does is writing a Date&Time stamp to the memory beginning at address 0000. Then the data follow.

This Date&Time stamp is repeated in intervals depending on the chosen saving mode:

- Mode 'CPS, save every second' — once every 10 min, or every 600 to 3000 bytes
- Mode 'CPM, save every minute' — once every hour, or every 60 to 300 bytes
- Mode 'CPM, save hourly average' — once every hour, or every dozen bytes. For unknown reasons the saving occurs exactly once every 1 hour + 8 … 13 seconds; this difference is ignored in GeigerLog.
- Mode 'OFF (no history saving)' — nothing is written; not even a message that saving was switched off

The wide ranges with respect to bytes result from the fact that a count rate (CPS or CPM) of up to 255 takes one byte to store, but a higher count rate takes 5 bytes, consisting of now 2 bytes of data, preceded by a 3 byte double-byte-announcing-tag! The 2 bytes now allow up to 65535 counts.

However, I noted an inconsistency in the readings of CPS double-byte data, which may be due to some undeclared use of the top two bits by the firmware. Therefore GeigerLog masks those two bits for CPS values, and therefore the maximum reading is 16383 counts. CPM might also be affected in the same way, but such a high reading has not been seen. Currently no CPM mask is effective.

Data bytes are saved at the end of the period following the Date&Time stamp. It does not matter much in the second and minute saving intervals, but in the hourly case it may matter.

The Date&Time stamp also carries the information of the saving mode. Without that you can't interpret the data, as it could have been saved every second, or every minute, or every hour, as CPS or as CPM! The saving mode is valid until the next Date&Time stamp.

If a Note/Location tag was entered at the Geiger counter device, then it will be stored after every Date&Time Stamp.

Overflow

Once the memory is filled, the bottom memory is prepared for the overflow by erasing the first page (a page = 4kB, 4096 bytes) of the memory. Again, erasing means overwriting with FF. Once this page is full, the 2nd page is erased, and so on.

The first issue to consider is that the time sequence in the memory from bottom to top is now: youngest data, followed by oldest data, which are becoming younger as you go up in memory. Therefore GeigerLog does a final sorting of all records according to time of each record determined by the parser.

Further, it is unlikely that the overflow begins with a Date&Time stamp at address 0000; instead the Date&Time stamp will come later within the regular flow of data. But since a Date&Time stamp is stringently required for the parsing, all data have to be skipped until a Date&Time stamp is found.

GeigerLog takes care of this missed overflow by linearizing the ring-buffer. Thereby those skipped data are attached to the top end of the memory copy, and will be parsed at the end.

Page Boundaries

Another issue is that deleting a page may cut through a tag, be it a Date&Time stamp, an ASCII tag, or a 5 byte double-data-byte-tag, making the left-over data uninterpretable or worse, giving them a totally different meaning. Following is an example, taken from an actual recording.

In the old recording a Date&Time stamp begins at byte index 4089 (in green; 2017-02-15 09:19:12, CPM saving every minute), and extends over the page boundary (P) into the second page. It is followed immediately by another Date&Time stamp at byte index 4101 (in blue; first 4 bytes only).

```
 4085:aa=170   4086:02=  2   4087:11= 17   4088:0e= 14 4089:55= 85
 4090:aa=170   4091:00=  0   4092:11= 17   4093:02=  2 4094:0f= 15
 4095:09=  9 P 4096:13= 19   4097:0c= 12   4098:55= 85 4099:aa=170
 4100:02=  2   4101:55= 85   4102:aa=170   4103:00=  0 4104:11= 17
```

After the page is deleted, all bytes up to the end of the page are set to 255 (in gray). The former time fragments 19 (min) and 12 (sec) become regular counts (in white) and the remainder of the Date&Time stamp beginning at 4098 (in yellow) has now become an ASCII tag with 85 bytes of supposed ASCII code following (only 3 bytes shown) [28)].

```
 4085:ff=255   4086:ff=255   4087:ff=255   4088:ff=255 4089:ff=255
 4090:ff=255   4091:ff=255   4092:ff=255   4093:ff=255 4094:ff=255
 4095:ff=255 P 4096:13= 19   4097:0c= 12   4098:55= 85 4099:aa=170
 4100:02=  2   4101:55= 85   4102:aa=170   4103:00=  0 4104:11= 17
```

28 Actually, as ASCII is limited to a 7 bit code, values of 128 and greater are not ASCII code; but GeigerLog is generous and reads it as an 8 bit code. It is nonsense anyway.

There is no way to put any meaning back into these fragments, therefore all data up the next Date&Time stamp must be discarded.

Another example from an actual recording: The Date&Time stamp (in yellow, 2022-02-04 05:48:19; ignore the date being 5 years into the future, this is yet another problem of the counter firmware) extends across a page boundary. The value at 28672 (in orange) is the Saving Mode byte, which can have values of 0, 1, 2, or 3. But it is 255.

```
28660:10=16   28661:55=85    28662:aa=170   28663:00=0     28664:16=22
28665:02=2    28666:04=4     28667:05=5     28668:30=48    28669:13=19
28670:55=85   28671:aa=170 P 28672:ff=255   28673:ff=255   28674:ff=255
```

The parser can only conclude that this is improper and all subsequent values until the next Date&Time stamp are made negative to mark illegitimate data. If you see negative counts – this is the reason.

The 255 value

How many of the value 255 bytes do you need to see in order to conclude that these stand for 'empty' bytes? If there are hundreds, it seems clear. But where do you set the limit? If there are only three, two, or just one, they might well be correct counts, leaving the parser no choice but to consider the next bytes as correct as well. Most of the time this is nonsense.

GeigerLog's default action is to ignore all single bytes with value 255! This results in an error when you measure counts near 255, be it CPS or CPM. Apart from changing the average, you will loose 1 second or minute, resp., in the time tag. But this is corrected with the next Date&Time stamp.

You can change this default action by starting GeigerLog with (see also menu Help → Options):

```
./geigerlog keepFF
```

This will result in all values 255 being treated as if they are correctly measured values. But most of the time this will be a mess, which needs to be corrected manually.

Correcting a Wrong History

It is an annoying procedure. The following is suggested:

1. Download the full history from the counter, and look at the graph
2. Try to zoom into the critical zone with mouse-left-click and mouse-right-click followed by Apply. Do it until you are able to read the time and count value of a relevant data point
3. Search the *.his file for this data point and note the byte index (first column)
4. Search the *.lst file for this byte index, and determine which data need to be deleted
5. Using a program able to handle binary files, delete the segment just determined in *.bin file
6. The remaining *.bin file can now be opened and parsed again, and should result in a proper history. If not, repeat at step 2.

Appendix F – Firmware Differences

The firmware of the GQ Geiger counters has bugs. Nobody is surprised that software has bugs. The unpleasant part is that GQ was not the most forthcoming in disclosing these bugs after they became known.

Furthermore, the firmware is modified from model to model. So far a normal process. Though what the modifications were, was not disclosed. Of course, it is completely up to the owner of this software to decide on what to publish or not, were it not for their simultaneous promotion and marketing of their products as 'open', as done for all models including their very latest GMC-600+, quote: "GQ GMC-600 Plus provides **open** GQ RFC1201 communication protocol for easier system integration". Well, no. This document had flaws at the time of release in Jan 2015 for the then latest GMC-300 models, and today has significant differences to the real situation, despite claim to the opposite. You surely can't do any "system integration" based on this outdated document.

I was therefore very pleased that GQ had decided to come forward with helpful information, which is mostly included in the extended online discussion in this post with topic 4948:
http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4948

This has allowed to fully integrate the 500 and 600 series into GeigerLog 0.9.07!

However, during this discussion some more firmware bugs surfaced. While they don't seem to impact the function of GeigerLog, you can never be sure about what is going on as long as you have not at least understood the issues, let alone haven't solved them.

- Both Logging and History download is working on all models
- Reading the calibration works on all series
- Reading and Setting Geiger counter configurations like, alarm, speaker, power status, History saving mode works for all series. However, they do not work reliably, not even for the old 300 series counters: every now and then a function fails, which always turned out to be due to an unexpected timeout of the counter. This is an issue of the counter's firmware! GeigerLog attempts to correct the failure, and is mostly, but not always successful. Look for the output printed to the NotePad. Your command may have not been successful; repeat the command if it did not succeed.
- If you find a problem, and can repeat it, you might want to start geigerlog with the Debug options, like:

  ```
  geigerlog -dvR
  ```

  This will result in a protocol file named `geigerlog.stdlog` which is needed for debugging. See Problems and Bugs for further handling.

History Download issues

The history is downloaded in pages of up to 4k (4096) bytes, which is hexadecimal 1000. The download is triggered by a request from the computer to send a page of the desired size. This desired size is then logically ANDed with hex0FFF, with the consequence that (hex1000 AND hex0FFF) = 0 – and hence no bytes are send by the Geiger counter at all!

Such is the situation with the ‘GMC-300 v3.20’ Geiger counter, which necessitates to limit the reading to half pages with a size of 2k.

In later models this firmware bug has been modified to a different firmware bug, whereby one byte more than requested is sent. When requesting a full 4k page of data, the firmware sends only (hex1000 AND 0FFF + 1) = 1 data point instead of 4096. The workaround is to request 4096 - 1 = 4095 data points, which results in (hex0FFF AND hex0FFF) = hex0FFF, then adds 1, resulting in hex1000, or, voilà, the full 4096 bytes.

Such is the situation with the ‘GMC-300E Plus v4.20’ and ‘GMC-320’ (assuming v4.20 firmware).

Note that this cannot be corrected by asking all counters for 2k half-pages only, as the extra byte send by the later firmwares still needs to be taken care off!

In the 500 and 600 series this extra-byte modification seems to have been reversed. I don’t know how, but reading only half pages (2k) is working.

Configuration Issues

For the 300 series the configuration is stored in a memory of 256 bytes. There is confusion around the meaning of each entry (see Appendix D – The GMC Device Configuration Meanings, page 63), though most is understood.

For the 500 and 600 series the configuration is twice as long at 512 bytes, and with the recent disclosure by GQ, the meaning is now defined (http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4948).

Double-tube Counters

The history for the 500+ counters now allows to save the sum of the counts of both tubes or either the first or the second tube.

The sum of both tubes does not make any sense at all, but is the current default setting of all firmware so far. The first tube is the more sensitive tube, typically a M4011 tube, and the second tube is the (much) less sensitive SI3BG tube.

The tube choice will be auto-detected from the downloaded history.

History Downloads using GQ Dataviewer and analyzing with GeigerLog

The GQ software Dataviewer may add the configuration memory of 256 or 512 bytes to the downloaded history memory, which may result in false parsing results. GeigerLog now tries to eliminate such wrong data.

Appendix G – Calibration

The calibration is meant to establish a relationship between the count rate in CPM and the dose rate in µSv/h. The GQ GMC counters have 3 calibration points, which would allow to accommodate some non-linearity in the relationship to take care of count rate saturation effects. However, currently all 3 points establish the same slope, hence effectively only a single calibration point is used [29]):

```
Device Calibration:
 Calibration Point 1:   60 CPM = 0.39 µSv/h  (0.0065 µSv/h / CPM)
 Calibration Point 2:  240 CPM = 1.56 µSv/h  (0.0065 µSv/h / CPM)
 Calibration Point 3: 1000 CPM = 6.50 µSv/h  (0.0065 µSv/h / CPM)
```

GeigerLog uses a default calibration of 0.0065 µSv/h / CPM except for the GMC-600+ where it uses 0.002637 µSv/h / CPM, but this can be changed in the configuration file.

Unfortunately, there is no official statement about what this actually means. For what situations is it applicable? What type of radioactivity? What beta, gamma energies? What count rates?

I was unable to find specifications for the Geiger counter tube M4011, currently used in GQ counters. However, I established that the SBM20 tube, an old Russian Geiger tube, is similar to the M4011 at least in some aspects, and can even be used instead of the M4011 in the GMC-300E+ counter, see http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4571 .

And for the SBM20 one does find specifications, like here: http://www.gstube.com/data/2398/

Gamma Sensitivity Ra^{226} (cps/mR/hr)	**29**
Gamma Sensitivity Co^{60} (cps/mR/hr)	**22**

Co60 is a beta and gamma emitter; Ra226 is an alpha, beta and gamma emitter. However, both are typically packaged such that only gamma can escape the package, and so we now assume pure gamma emission. With that we can equate mR with mRem, and with 1 mRem = 10µSv, we get:

Ra226: 29 * 60 / 10 = 174 CPM / (µSv/h); invers: → 0.0058 (µSv/h) / CPM
Co60: 22 * 60 / 10 = 132 CPM / (µSv/h); invers: → 0.0076 (µSv/h) / CPM
Average of the two: 0.0067 (µSv/h) / CPM
GQ's calibration: 0.0065 (µSv/h) / CPM

GQ's calibration is close enough to the average of the two, and with nothing better at hand we'd say that this is the same, and that this is the base for GQ's calibration factor.

Looking at the gamma spectra in fig. 10 we see that Co60 is above 1 MeV, while Ra226 is mostly below 0.5 MeV. At least the SBM20 tube, according to specs, is 32% more sensitive to the lower

29 Careful when you use a GMC-500: they were delivered with this calibration setting

```
Device Calibration:
   Calibration Point 1:            60 CPM =   0.39 µSv/h  (0.0065 µSv/h / CPM)
   Calibration Point 2:         10000 CPM =  65.00 µSv/h  (0.0065 µSv/h / CPM)
   Calibration Point 3:            25 CPM =   9.75 µSv/h  (0.3900 µSv/h / CPM)
```

In recent comments by GQ this was attributed to a GMC-500+ device, and it was explained that this handles the second tube in this device. However, this calibration was also found in a GMC-500, which has not second tube.

energy gammas. Perhaps because the higher energy gammas of Co60 have a lower absorption and hence a better chance to pass through the tube without generating a count.

The consequence is that the calibration is ONLY applicable for gamma radiation (and only approximately given the energy dependence), but NOT for beta radiation, for which both tubes are also sensitive!

We simply do not know what the calibration factor is for beta!

Since the case of the counter is basically transparent to gammas, it does not matter to the calibration whether we make the backplate of the counter more permeable by drilling holes, or taking the backplate off completely – especially considering the hand waving we have applied to come up with the gamma calibration.

And when we take it off and get significantly higher count rates with beta emitters, it also does not matter because the calibration, when applied to beta, is wrong in the first place!

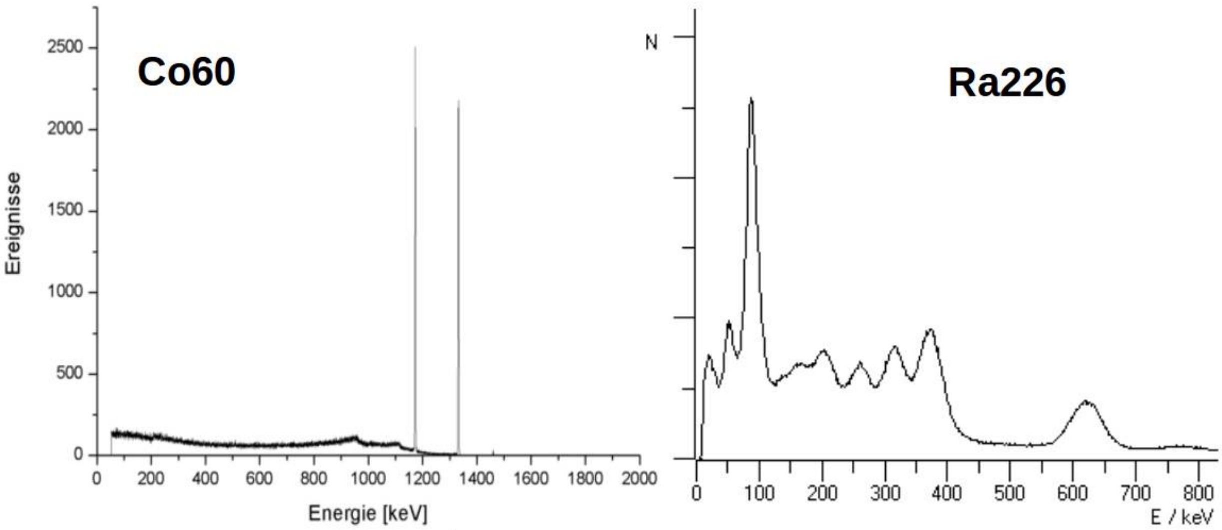


Fig. G1 Gamma Spectra of Ra226 and Co60

Appendix H – Installation

A full working environment for GeigerLog needs the Python interpreter and some supporting packages matching the installed version of Python.

If these conditions are met, GeigerLog will run on **Linux**, **Windows**, **Mac**, and **other systems**!

Python

This version of GeigerLog requires **Python version 3.x**. It has been developed on Python 3.5.2, and was confirmed to also work on Python 3.4, 3.6, and 3.7. The latest Python is currently (Aug 2019) version 3.7.4.

Pip

For any Python installation – be it on Linux, Windows, Mac, or else – it will be almost impossible to get a fully working installation without having the program **Pip** [30]) also installed!

All of the above Python versions come packaged with Pip, and while Pip should normally be installed by default, make absolutely sure that it is! Note that there might be a checkbox in the Python installer, which **must be checked in order to install Pip!**

64 bit versus 32 bit

Use a 64 bit installation if your operating system supports it. The following assumes 64 bit for all downloads. If you have to use 32 bit, find the equivalent downloads.

Administrative rights

Whenever you have the choice, install **with administrative rights**!

Verify the current Python and Pip installation status on your machine

On your machine you may have installed only Python version 2.x (Py2), or only Python version 3.x (Py3), or both, or neither. Furthermore, depending on your operating system and distribution, as well as your history of installations, one of the two can probably be started with ‘python’, while the other needs to be started with ‘python2’ or ‘pathon3’.

Likewise with pip.

To find out your situation, look at the output of these commands entered in a Command Window:

```
python -V
python2 -V
python3 -V
pip -V
pip2 -V
pip3 -V
```

30 Pip is a recursive acronym that can stand for either "Pip Installs Packages" or "Pip Installs Python". Homepage: https://pypi.org/project/pip/

The responses will tell you which command starts which version of Python and Pip.

We will now assume that both Python and Pip are installed as version 3, and you have to use the commands '**python3**' and '**pip3**' to start your Python 3 and matching Pip. If your installation requires different commands, use those instead in the following statements.

If you have no Python 3 installed, look below for installation instructions specific for your operating system.

Other Packages

In addition to a working Python installation, you also need certain Python packages, which you probably won't have in a default installation. How to install them will be explained in the next chapters.

Latest versions as of Aug 2019:

• PyQt5	latest version: 5.13.0
• PyQt5-sip	latest version: 4.19.18
• pip	latest version: 19.2.1
• setuptools	latest version: 41.0.1
• matplotlib **(MUST NOT be version 3.1.0)**	latest version: 3.0.3
• numpy (**MUST be at least version 1.14**)	latest version: 1.17.0
• scipy	latest version: 1.3.1
• pyserial (**MUST be at least version 3.4**)	latest version: 3.4
• paho-mqtt	latest version: 1.4.0
• pyaudio (**MUST be at least version 0.2.11**)	latest version: 0.2.11

While for most packages a 'recent' version should suffice, note the exceptions! For matplotlib see the later explanations under **Installation Problems** for your operating system!

Installing Using Pip

Not all packages may be installable with Pip, but if they are, you should prefer Pip over the installation tool of your operating system or distribution! Pip will also give you a more recent version. Some advanced use of Pip is explained in chapter Appendix I – Advanced Use of Pip on page 85, here some brief basic usage, as an example for the installation of numpy:

To do a fresh install of package numpy, simply do from a Command Window:

```
pip3 install numpy
```

If numpy is already installed, verify that the version is at least 1.14:

```
pip3 show numpy
```

If an upgrade is needed, do it with:

```
pip3 install --upgrade numpy
```

(NOTE: **2** dashes before upgrade!)

To list all installed Python modules with their version, use:

```
pip3 list
```

What if all installed well but GeigerLog fails to run?

To get more information on the problem from GeigerLog, start it from a Command Window with the options debug ‘-d’ and verbose ‘-v’ :

```
geigerlog  -dv
```

You’ll find its output in the terminal and in the program log file `geigerlog.proglog`. Look through these messages to find out what went wrong.

Perhaps some modules are missing? Modules may simply be not installed, but may be installed, though in a deprecated version.

Try to re-install and update these modules using Pip.

Sometimes, however, there is a conflict when the module installed by the distribution is too old, and does not allow to be updated by Pip. If Pip complains that it can’t do an update, then un-install this package first with the distribution tools, like for Ubuntu:

```
apt-get purge <package-name>
```

and only then reinstall with Pip. (Such was the case for the pyserial module in the Mint distribution.)

Still not found the problem?

Start GeigerLog with the options debug ‘-d’ and verbose ‘-v’ and very verbose ‘-w’ and Redirect ‘-R’, like:

```
geigerlog  -dvwR
```

This will redirect all output – including error messages of the operating system – to a file in the data directory named `geigerlog.stdlog`. Bring this to my attention via the Sourceforge site of GeigerLog: https://sourceforge.net/projects/geigerlog/ .

Installation of an Editor

Even if you don’t want to edit the program code, you need a proper editor to adapt e.g. the GeigerLog configuration file to your needs, without messing with the line endings, which some Windows editors like to do.

GeigerLog has been developed on the editor Geany, which I do recommend. If you don’t have a good editor yet, consider installing it. Latest release is Geany 1.35 (Aug 2019).

Get Geany from: https://www.geany.org/Download/Releases .

Linux – Installation

This was tested with **Ubuntu Mate 16.04.6 LTS, 18.04 LTS,** and **19.04** using only those repositories present in the default installation.

Installation of Python and Pip

```
sudo apt-get install python3
sudo apt-get install python3-pip
```

Updating Pip and more Installations

Once Pip for Py3 is installed, use it to upgrade itself, and **only then** install the other packages.

A ‘sudo -H’ is required for administrative installation. The `‘--upgrade’` (NOTE: 2 dashes before ‘upgrade’!) at each command ensures that the most recent version of each module will be installed even if an older version is already installed (tip: copy&paste the whole next paragraph to a terminal):

```
sudo -H pip3 install --upgrade pip
sudo -H pip3 install --upgrade setuptools
sudo -H pip3 install --upgrade pyqt5
sudo -H pip3 install --upgrade pyqt5-sip
sudo -H pip3 install --upgrade matplotlib
sudo -H pip3 install --upgrade numpy
sudo -H pip3 install --upgrade scipy
sudo -H pip3 install --upgrade pyserial
sudo -H pip3 install --upgrade paho-mqtt
sudo -H pip3 install --upgrade pyaudio
```

List all installations and verify that versions of numpy, pyserial, and pyaudio meet the minimum requirements (see page 76 for Other Packages)

```
pip3 list
```

Installation of GeigerLog

Copy the `geigerlog-scripts-vXYZ.zip` file to a directory of your choice and unpack. The unpacking will have created the folder `geigerlog` with the required content.

Change to the `geigerlog` folder and start GeigerLog from the terminal with:

```
./geigerlog
```

or create a starter to be put in the panel.

Installation Problems

Note: Special thanks to user theMike!

Pyaudio:

Installation failed because **python3-dev** was missing. Install with the distribution's installer, like **'sudo apt-get install python3-dev'** on Ubuntu.

Installation failed because **portaudio19-dev** was missing. Install with the distribution's installer, like **'sudo apt-get install portaudio19-dev'** on Ubuntu.

GeigerLog requires a minimum **version 0.2.11** for pyaudio.

Matplotlib:

A version 3.1.0 had been released, which had a severe bug and would not work at all in GeigerLog. This version apparently has been withdrawn and can no longer be installed! However, if you have any such installation, force the installation of the previous version 3.0.3, which works well:

```
sudo -H pip3 install matplotlib==3.0.3
```

if this fails, try:

```
sudo -H pip3 install --force-reinstall matplotlib==3.0.3
```

Numpy, Scipy

If you encounter any issues with these two packages, you might test them. First you probably have to install pytest:

```
sudo -H pip3 install --upgrade pytest
```

then run

```
scipy.test(label='full', verbose=2)

numpy.test(label='full', verbose=2)
```

For referenes see: https://stackoverflow.com/questions/9200727/is-there-a-test-suite-for-numpy-scipy/9200923#9200923 and https://docs.scipy.org/doc/numpy-1.15.0/reference/testing.html .

Windows - Installation

This was tested with an installation of **Windows 10 Pro** on two different computer.

Existing Installation with Old Version of Python

If you already have **GeigerLog version 0.9.07 or later** running on any version of **Python 3**, **AND** you are able to install the latest packages specific for this new GeigerLog version, you don't have to upgrade Python. Give it a try:

Open a Command Window **as administrative user**. Issue the commands (see above for the question of using pip or pip3 as command):

```
pip3 install --upgrade pip
```

```
pip3 install --upgrade setuptools
pip3 install --upgrade pyqt5
pip3 install --upgrade pyqt5-sip
pip3 install --upgrade matplotlib
pip3 install --upgrade numpy
pip3 install --upgrade scipy
pip3 install --upgrade pyserial
pip3 install --upgrade paho-mqtt
pip3 install --upgrade pyaudio
```

List all installations and verify that versions of numpy, pyserial, and pyaudio meet the minimum requirements (see page 76 for Other Packages):

```
pip3 list
```

If all successful, you can skip the next chapter and go to Installation of GeigerLog. If not, follow the next chapter for a new install of Python.

New version Python 3.7.2

You will install Python version 3.7.2, which is the very latest of the series, released Dec. 24, 2018 (status March 2019).

Cleanup !

Previous installations of Python may have scattered files all around the hard disks, and this may get severely in the way of a new install. Therefore a thorough cleanup is very strongly recommended!

First, delete your current Python 3 installation using Python's installer.

Second, remove all remnants in the Windows-Registry. I used the freeware version of CCleaner (https://www.ccleaner.com/ccleaner) for it.

Third, you need to hunt for any Python leftovers, and delete it. In particular look under '…\<username>\AppData\Roaming\...'.

Installation of Python

Download the latest version of Python from: https://www.python.org/downloads/windows/.

You want the "**Windows x86-64 executable installer**" :
https://www.python.org/ftp/python/3.7.2/**python-3.7.2-amd64.exe**

Open it **with administrative rights!**

In the installer:

- check: add Python 3.7 to Path
- select: Customize Installation
- under Optional Features:
  - check all options
  - select Next

- under Advanced Option:
  - check all options (except for the last two 'Download…' items; they are not needed)
  - leave the install path at 'C:\Program Files\Python37'
  - select Install
- once finished: ignore the option to disable path length limit and close the installer

Updating Pip and more Installations

If your Command Prompt window is still open, then close it (Really!). Open again **as administrative user**.

Remember to verify (see chapter Verify the current Python and Pip installation status on your machine on page 75) what your command is to call Pip in version 3! I continue to use 'pip3' as a reminder to use the proper version, although if your system has no other Python version aboard except for the just installed one, your command might be only 'pip'. Type:

```
pip3 list
```

You will see only pip and setuptools listed (unless you have an additional Python installation) and likely both versions are outdated. Upgrade them and install all others with these commands (The '`--upgrade`' at each command ensures that the most recent version of each module will be installed even if an older version is already installed):

```
pip3 install --upgrade pip
pip3 install --upgrade setuptools
pip3 install --upgrade pyqt5
pip3 install --upgrade pyqt5-sip
pip3 install --upgrade matplotlib
pip3 install --upgrade numpy
pip3 install --upgrade scipy
pip3 install --upgrade pyserial
pip3 install --upgrade paho-mqtt
pip3 install --upgrade pyaudio
```

List all installations and verify that versions of numpy, pyserial, and pyaudio meet the minimum requirements (see page 76 for Other Packages):

```
pip3 list
```

Installation of GeigerLog

It is suggested to place GeigerLog directly under `c:\`. Unzip the content of `geigerlog-scripts-vXYZ.zip` file to c:\. This will have created the folder `c:\geigerlog` with the required content.

Start GeigerLog from a Command Prompt window with:

```
python3 c:\geigerlog\geigerlog
```

More conveniently, create a shortcut to the file `geigerlog` in your geigerlog folder and place the shortcut on your desktop. Then open the shortcut's properties and change its Target to:

```
python3 c:\geigerlog\geigerlog
```

Every time you click the shortcut, a Command Prompt window will open and GeigerLog will be started from there. Output from GeigerLog will go into this window, but in addition always also into the program log file `geigerlog.proglog`.

If you don't want this extra Command Prompt window, then edit the shortcut's Target to:

```
python3w c:\geigerlog\geigerlog
```

Remember to replace '`python3`' with whatever your system requires; the '`python3w`' might be a simple '`pythonw`'!

Installation Problems

Windows specific issues

When you encounter error messages like:

```
ImportError: DLL load failed: The specified module could not be found.
```

The most likely reason is that "Microsoft Visual C++ Redistributable" is missing. Install from the Micorsoft website:

https://support.microsoft.com/ms-my/help/2977003/the-latest-supported-visual-c-downloads

Pyaudio:

Note: Special thanks to user ikerrg!

The pyaudio website has no supporting files precompiled for the use with pip, and compilation options are typically not available for standard Windows installations. As a consequence the installation of pyaudio will fail with the standard pip command.

Workaround:

Download a so called 'Wheel' file from here: https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyaudio and install.

Example for a 64bit Windows:

Download **PyAudio-0.2.11-cp37-cp37m-win_amd64.whl**, change into the directory where the download was saved, and install with administrator privileges with command:

```
pip3 install PyAudio-0.2.11-cp37-cp37m-win_amd64.whl
```

GeigerLog requires a minimum **version 0.2.11** for pyaudio.

Matplotlib:

A version 3.1.0 had been released, which had a severe bug and would not work at all in GeigerLog. This version apparently has been withdrawn and can no longer be installed! However, if you have any such installation, force the installation of the previous version 3.0.3, which works well:

```
pip3 install matplotlib==3.0.3
```

if this fails, try:

```
pip3 install --force-reinstall matplotlib==3.0.3
```

Numpy, Scipy

If you encounter any issues with these two packages, you might test them. First you probably have to install pytest:

```
pip3 install --upgrade pytest
```

then run

```
scipy.test(label='full', verbose=2)

numpy.test(label='full', verbose=2)
```

For referenes see: https://stackoverflow.com/questions/9200727/is-there-a-test-suite-for-numpy-scipy/9200923#9200923 and https://docs.scipy.org/doc/numpy-1.15.0/reference/testing.html .

Mac – Installation

The following has not been tested on a Mac, but is derived from various online sources. A HOWTO for using Python on a Mac is available on this site from the Python creators: https://docs.python.org/3/using/mac.html and covers relevant topics .

To install Py3 see instructions under the above link. A "universal binary" build of Python, which runs natively on the Mac's new Intel and legacy PPC CPU's, is there available.

Note the caveat on starting programs with a GUI (Graphical User Interface, which GeigerLog has) due to a quirk in Mac.

The latest Python releases for Mac are here: https://www.python.org/downloads/mac-osx/ Latest version currently is 3.6.5. Choose your installer for 64bit-only (preferred if possible) or 32/64 bit.

Now verify the Python installation:

```
python3 -V                    → expected: =3.6.5
```

Now with Python working, verify your installation status on your machine with the commands given in chapter Verify the current Python and Pip installation status on your machine on page 75.

Using pip

Once pip for Py3 is installed, upgrade it and install all others (The '`--upgrade`' (NOTE: **2** dashes before 'upgrade'!) at each command ensures that the most recent version of each module will be installed even if an older version is already installed):

```
pip3 install --upgrade pip
pip3 install --upgrade setuptools
pip3 install --upgrade pyqt5
pip3 install --upgrade pyqt5-sip
pip3 install --upgrade matplotlib
pip3 install --upgrade numpy
pip3 install --upgrade scipy
pip3 install --upgrade pyserial
pip3 install --upgrade paho-mqtt
pip3 install --upgrade pyaudio
```

List all installations and verify that versions of numpy, pyserial, and pyaudio meet the minimum requirements (see page 76 for Other Packages):

```
pip3 list
```

Installation of GeigerLog

Copy the `geigerlog-scripts-vXYZ.zip` file to a directory of your choice and unpack. The unpacking will have created the folder `geigerlog` with the required content.

Start GeigerLog from the terminal with:

```
geigerlog
```

Installation Problems

Matplotlib:

A version 3.1.0 had been released, which had a severe bug and would not work at all in GeigerLog. This version apparently has been withdrawn and can no longer be installed! However, if you have any such installation, force the installation of the previous version 3.0.3, which works well:

```
pip3 install matplotlib==3.0.3
```

if this fails, try:

```
pip3 install --force-reinstall matplotlib==3.0.3
```

Appendix I – Advanced Use of Pip

More on Pip is here: https://pip.pypa.io/en/stable/ . Some helpful commands are:

List all versions for which an update is available:

```
pip3 list --outdated
```

Installing a specific version

To install a specific python package version irrespective whether it is for the first time, for an upgrade or a downgrade, use (e.g. for the package `paho-mqtt`):

```
pip3 install --force-reinstall paho-mqtt==1.3.1
```

Looking for a specific version

To view all available package versions from an index exclude the version number, like:

```
pip3 install paho-mqtt==
```

which will result in an error message like:

```
ERROR: Could not find a version that satisfies the requirement paho-mqtt== (from versions: 0.4.90, 0.4.91, 0.4.92, 0.4.94, 0.9, 0.9.1, 1.0, 1.1, 1.2, 1.2.1, 1.2.2, 1.2.3, 1.3.0, 1.3.1, 1.4.0)
ERROR: No matching distribution found for paho-mqtt==
```

thereby giving you the available versions.

Helpful Tools: Pip-check

Pip-check gives you a conveniently formatted overview of all installed packages and their update status. The homepage is https://pypi.org/project/pip-check/ .

Note: there is also a package ‘pipcheck’ (without a dash in the name), which is the wrong one! The right one is ‘**pip-check**’, version 2.5.2 (Aug 2019).

Install or update pip-check:

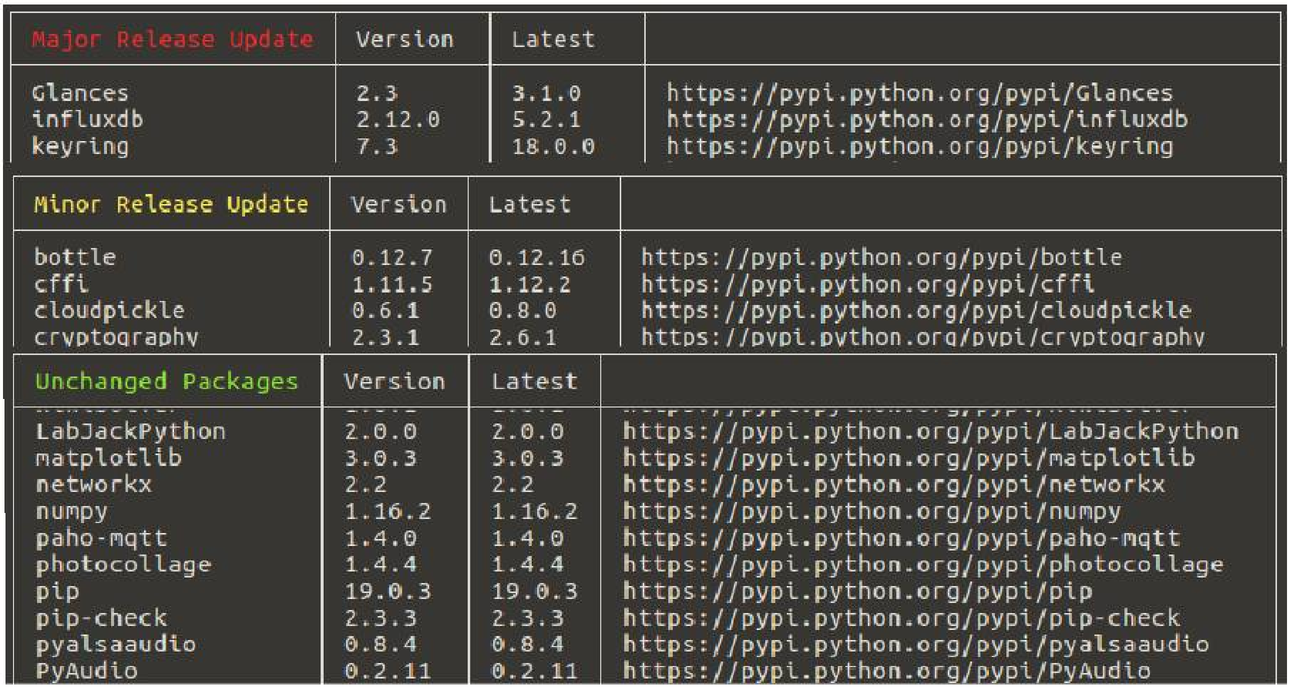
```
pip3 install --upgrade pip-check
```

Run pip-check:

```
pip-check
```

Run pip-check when on your system Pip needs to be called not just as pip but as pip3:

```
pip-check  --cmd pip3
```



Major Release Update	Version	Latest	
Glances	2.3	3.1.0	https://pypi.python.org/pypi/Glances
influxdb	2.12.0	5.2.1	https://pypi.python.org/pypi/influxdb
keyring	7.3	18.0.0	https://pypi.python.org/pypi/keyring

Minor Release Update	Version	Latest	
bottle	0.12.7	0.12.16	https://pypi.python.org/pypi/bottle
cffi	1.11.5	1.12.2	https://pypi.python.org/pypi/cffi
cloudpickle	0.6.1	0.8.0	https://pypi.python.org/pypi/cloudpickle
cryptography	2.3.1	2.6.1	https://pypi.python.org/pypi/cryptography

Unchanged Packages	Version	Latest	
LabJackPython	2.0.0	2.0.0	https://pypi.python.org/pypi/LabJackPython
matplotlib	3.0.3	3.0.3	https://pypi.python.org/pypi/matplotlib
networkx	2.2	2.2	https://pypi.python.org/pypi/networkx
numpy	1.16.2	1.16.2	https://pypi.python.org/pypi/numpy
paho-mqtt	1.4.0	1.4.0	https://pypi.python.org/pypi/paho-mqtt
photocollage	1.4.4	1.4.4	https://pypi.python.org/pypi/photocollage
pip	19.0.3	19.0.3	https://pypi.python.org/pypi/pip
pip-check	2.3.3	2.3.3	https://pypi.python.org/pypi/pip-check
pyalsaaudio	0.8.4	0.8.4	https://pypi.python.org/pypi/pyalsaaudio
PyAudio	0.2.11	0.2.11	https://pypi.python.org/pypi/PyAudio

Fig. I1 Output of pip-check in a Terminal

Appendix J – License

GeigerLog is licensed under GPL3. The license text is available in file `COPYING` in the GeigerLog folder. If the file is missing you find a link to it in this text, which is part of all GeigerLog files:

```
###########################################################################
#    This file is part of GeigerLog.
#
#    GeigerLog is free software: you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation, either version 3 of the License, or
#    (at your option) any later version.
#
#    GeigerLog is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public License
#    along with GeigerLog.  If not, see <http://www.gnu.org/licenses/>.
###########################################################################
```