



# **GeigerLog Manual**

by ullix

Version 1.0

September 2020

---

## What's New in GeigerLog 1.0 ?

Short answer: very little! That's why it became the 1.0 release.

- Installation has again become a little easier, in particular on Windows.
- A guidance for the installation of GeigerLog on a Raspberry Pi has been added to this manual, including a use example with a GMC-300E+ counter
- One thing, however, has gotten a major revision, which is the **Calibration Factor**. It is now the **inverse** of the old definition:

$$\text{new Calibration Factor} = 1 / \text{old Calibration Factor}$$

Why the change? To make things easier. Read more in chapter Appendix G – Calibration on page 92.

## 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 show how to best use it, taking advantage of GeigerLog on 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.

### [GeigerLog - Radiation-v1.1\(CAJOE\)-Support](#)

The **Radiation-v1.1 (CAJOE)** Geiger counter – called **Cajoe-Counter** for short – is a low cost Geiger counter, which can only generate audio-clicks for each registered radioactive event. The article gives some details on the counter which are not easily available, and shows how to connect GeigerLog to it.

### [GeigerLog-HOWTO use Python in a virtual Environment on Linux](#)

When there is a need to use multiple versions of Python, or multiple configurations, you need to Python in its own virtual environment. Can be done surprisingly easily. Here shown for Linux.

Author	ullix
Credits	Phil Gillaspy for <a href="#">extended documentation</a> of Geiger counter commands GQ Electronics LLC for documentation
Copyright	Copyright 2016, 2017, 2018, 2019, 2020
License	GPL3, see also Appendix J – License on page 112 You should have received a copy of the GNU General Public License along with GeigerLog. If not, see <a href="http://www.gnu.org/licenses/">http://www.gnu.org/licenses/</a>

## Table of Contents

Overview.....	8
Introduction to GeigerLog.....	11
Installing and Starting GeigerLog.....	11
The GeigerLog Window.....	12
Quick Tour of GeigerLog.....	13
Running GeigerLog.....	14
Establishing a GeigerLog Connection to Your Devices.....	14
Device Mappings.....	15
Logging with GeigerLog.....	16
Manage Your Recordings – Data Dashboard.....	18
Visualize Your Recordings – Graph Dashboard.....	20
Configuration of GeigerLog.....	24
Loading *.log, *.his and Other CSV Files.....	31
Saving data as CSV formatted *.log, *.his Files.....	31
Quality Control of your Data.....	32
SuSt – Summary Statistics.....	32
Stats – Statistics.....	33
Poiss – Histogram with Poisson Fit.....	34
FFT – FFT & Autocorrelation Analysis.....	35
Scatter Plot.....	37
Printing Values of the Variables.....	40
Device-specific Considerations.....	41
GMC-Devices.....	41
AudioCounter Devices.....	46
I2C Devices.....	47
RadMon Devices.....	48
LabJack Devices.....	51
Gamma-Scout Devices.....	52
The GUI – Graphical User Interface.....	54
Menus.....	54
Toolbars.....	63
Miscellaneous.....	65
Starting GeigerLog with Options.....	65
Helpful Internal Software Tools.....	66
Helpful External Software Tools.....	69
None is required to run GeigerLog, but sometimes certain tools can be really helpful.....	69
Radiation World Maps.....	71
Occupational Radiation Limits.....	72
Problems and Bugs.....	73
References.....	73

Appendix A – Look & Feel.....	74
Appendix B – Connecting Device and Computer using a Serial Connection.....	75
Appendix C – HOWTO deal with read and write permissions for the serial port when on Linux....	80
Appendix D – The GMC Device Configuration Meanings.....	82
Appendix E – GMC Device: Internal Memory, Storage Format and Parsing Strategy.....	87
Appendix F – Firmware Differences.....	90
Appendix G – Calibration.....	92
Appendix H – Installation.....	100
Linux – Installation.....	103
Windows - Installation.....	105
Mac – Installation.....	107
Raspberry Pi Installations.....	108
Appendix I – Advanced Use of Pip.....	110
Appendix J – License.....	112

## Table of Figures

Figure 1: GeigerLog Window with Annotations.....	12
Figure 2: The Device Toolbar signaling the Connection Status.....	14
Figure 3: The Logging Toolbar’s various stages:.....	16
Figure 4: Data Dashboard.....	18
Figure 5: Set Log Timings.....	18
Figure 6: Set History Saving Mode.....	18
Figure 7: Graph Dashboard.....	20
Figure 8: The Display Last Log Values Pop-up Window.....	23
Figure 9: Startup Error on missing or non-readable configuration file.....	24
Figure 10: Startup Error on improper definitions in configuration file.....	24
Figure 11: Set Temporary Calibrations dialog.....	26
Figure 12: View and Edit Current Scaling Dialog.....	29
Figure 13: The Get Data from CSV File dialogue.....	31
Figure 14: Histogram of Low Count Rate.....	34
Figure 15: Histogram of High Count Rate.....	34
Figure 16: FFT Analysis of Medium Count Rate Measurement.....	35
Figure 17: Demonstration of the impact of convolution on an FFT spectrum.....	36
Figure 18: CPM and Temperature in a Temperature Experiment with a Geiger Counter.....	37
Figure 19: Select Variables for Scatter Plot.....	38
Figure 20: Scatter Plot with 2nd Order Polynomial Fit.....	38
Figure 21: Scatter Plot with 7th Order Polynomial Fit.....	38
Figure 22: Scatter Plot of CPM3rd=Audio Signal versus CPM=Digital Signal.....	39

Figure 23: Command File --> Show Plot Data showing only the data currently in the plot.....	40
Figure 24: GMC-300E+ (blue) digital recording, and SGP-001 Device (green) audio recording....	46
Figure 25: A Long-Term recording Using I2C-Devices.....	47
Figure 26: Activating RadMon Configuration on a Smartphone.....	49
Figure 27: RadMon+ Configuration pages.....	50
Figure 28: Long-Term Recording from an Outside RadMon+ Weather Station.....	50
Figure 29: Long-Term Recording – 2 years – with the LabJack, showing Temperature only.....	51
Figure 30: A recording from a Gamma-Scout device.....	53
Figure 31: The toolbars.....	63
Figure 32: Output from the Glsoundcheck for an audio source with negative pulses.....	67
Figure 33: Editor Geany with file geigerlog.cfg opened.....	69
Figure 34: DB Browser for SQLite as tool to inspect Log files.....	70
Figure 35: Show & Select Dialog for USB Port and Baudrate (result for a Linux system).....	77
Figure 36: Autodiscovery USB-to-Serial Port Connection.....	77
Figure 37: USB Autodiscovery result.....	78
Figure 38: Comparison of a M4011 and SBM20 tube inserted in a GMC-300E+ counter measuring Background.....	95
Figure 39: Comparison of a M4011 and SBM20 tube inserted in a GMC-300E+ counter measuring a Potassium and Thorium source, resp.....	95
Figure 40: Relative Response of a Geiger Tube with Respect to Gamma Energy.....	96
Figure 41: Synchrotron radiation captured with a GMC-500+ counter, and analyzed with Geiger-Log.....	98
Figure 42: Using a Raspberry Pi with GeigerLog.....	109
Figure 43: Output of pip-check in a Terminal.....	111

# 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 <sup>1)</sup> **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.

### 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, like the recently reviewed **Radiation-v1.1(CAJOE)** counter ([\(GeigerLog-Radiation-v1.1\(CAJOE\)\)-Support-v1.0.pdf](#)). Such counters can now be used with GeigerLog.

An audio connection also provides an alternative way to connect to 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)** ([\(GeigerLog-Review Smart Geiger Pro \(SGP-001\)\)](#)) semiconductor Geiger counter with a Personal Computer, which has so far not been possible!

### RadMon Devices:

GeigerLog supports the **RadMon+** <sup>2)</sup> 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

### 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/> ).

---

1 GQ Electronics LLC, 5608 Delridge Way SW, Seattle, WA 98106, USA, <http://www.gqelectronicsllc.com/>

2 DIYGeigerCounter <https://sites.google.com/site/diygeigercounter/>



### LabJack Devices:

GeigerLog supports the **Labjack** ( <https://labjack.com/> ) hardware U3 in combination with the ei1050 probe for temperature and humidity.

### Gamma Scout Devices:

GeigerLog fully supports the **Gama-Scout** (<https://www.gamma-scout.com/en/>) devices **Standard**, **Alert**, **Rechargeable**, and partially supports model **Online** (see Gamma-Scout Devices on page 52) .

### AmbioMon Devices:

GeigerLog supports the **AmbioMon**<sup>3</sup>) hardware, which can provide a Geiger counter as well as an environmental sensor for temperature, air-pressure (atmospheric-pressure), humidity, and air-quality, driven by an ESP32 microprocessor.

These devices can be controlled via smartphone, and transmit their data wirelessly.

## 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  $\mu\text{Sv/h}$ .
- The **right Y-axis** is reserved for environmental data. If temperature data are shown, the choice of units is between  $^{\circ}\text{C}$  and  $^{\circ}\text{F}$ .

To display variables with very different numerical values on a common scale – like temperature (e.g. 0 ... 30 $^{\circ}\text{C}$ ) 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, ...).

---

3 This device is in development and not yet publicly available

**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 as well as 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 <sup>4)</sup> !

It was developed with Python version 3.5.2, and verified to run with Python 3.4, 3.5, 3.6, 3.7, and 3.8 <sup>5)</sup>. Python 3.8 is now considered the version of choice!

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.

Step-by-Step installation instructions for Python on **Linux, Windows and Mac** are provided in Appendix H – Installation beginning on page 100.

### 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 itself 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 100.

### 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 74.

### Default Configuration

GeigerLog's default configuration is to use the GMC counters and the AudioCounters at the same time. This can be changed in the configuration file `geigerlog.cfg` (more on this in chapter Configuration of GeigerLog on page 24).

---

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!

5 As of September 2020 GeigerLog does not run on Python 3.9, because this Python version unfortunately does not yet support the essential component matplotlib!

# The GeigerLog Window

GeigerLog has a single window with predefined usage areas. Figure 1 gives an overview.

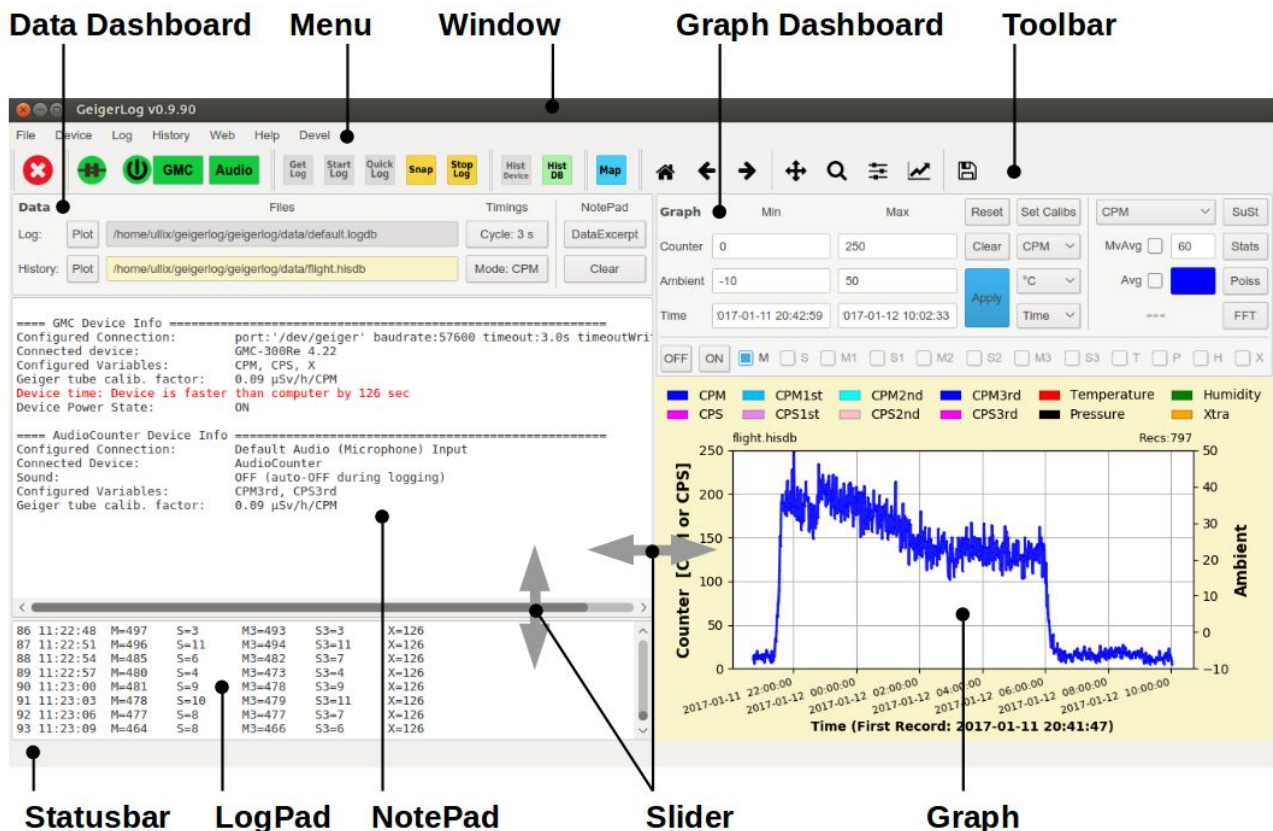


Figure 1: GeigerLog Window with Annotations

**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 (on a red background) 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 ' $\mu\text{Sv/h}$ '. The graph changes, now showing  $\mu\text{Sv/h}$  versus Time-of-Day. Now select the Time Unit drop-down button currently showing 'Time' and select 'auto'. The graph switches to  $\mu\text{Sv/h}$  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-framed-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; specific devices will be discussed later.

## 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 Figure 2.



Figure 2: The Device Toolbar signaling the Connection Status

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 brought down intentionally)

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
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
```

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 series, one of the Gamma-Scout counters series, and the AmbioMon++ series are devices, which – beyond a logging mode – support another operating mode: **History**. This means reading the data they have stored in their internal memory. It will be explained in a later chapter specific to these 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 18).

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



Figure 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) <sup>6</sup>) 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 35.

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!

---

6 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](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, and strength of the High-Voltage generator for the Anode voltage of the Geiger tube, lower this even further.

# Manage Your Recordings – Data Dashboard

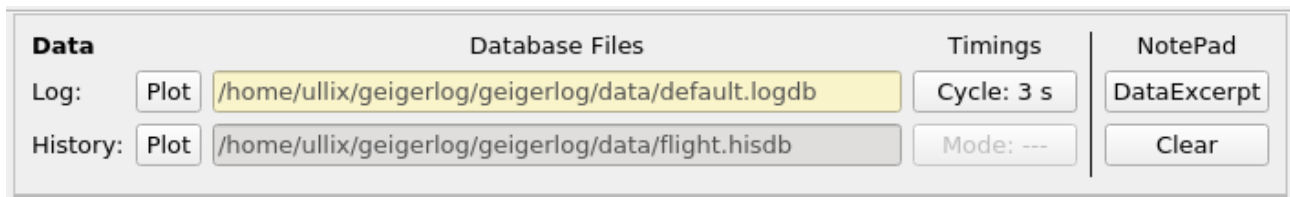


Figure 4: Data Dashboard

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.

## 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.

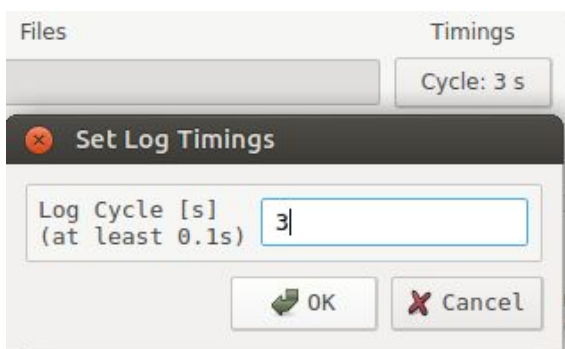


Figure 5: Set Log Timings

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.

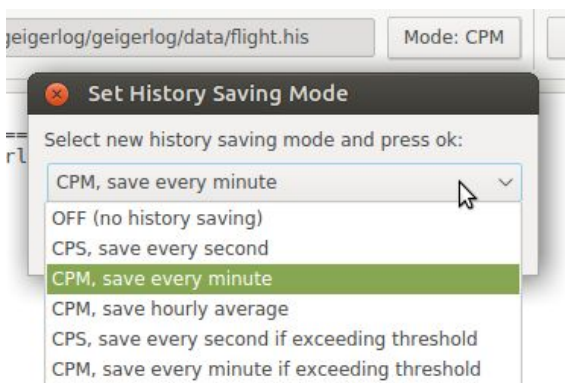


Figure 6: Set History Saving Mode

Clicking the **Mode** button opens a dialog box allowing you to choose between the available History Saving Modes for the GMC devices.

For an explanation of the options see History Saving Mode on page 44.

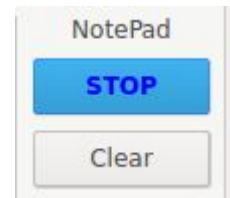
## NotePad

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 **DataExcerpt** button has a second function: when you print a full file via 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.



At any time the current content of the NotePad can also be printed to a printer, and saved to a file. Use the functions in menu File, **File** → **Print NotePad** and **File** → **Save NotePad to File**.

# Visualize Your Recordings – Graph Dashboard

The screenshot shows the Graph Dashboard interface. It includes input fields for Counter (0 to 700), Ambient (-10 to 50), and Time (9-09-04 10:48:50 to 9-09-04 12:16:57). There are buttons for Reset, Scaling, Clear, and Apply. A dropdown menu shows CPM, and a large yellow box displays 511.00 CPM. At the bottom, there are checkboxes for various variables: OFF, ON, M, S, M1, S1, M2, S2, M3, S3, T, P, H, X.

Figure 7: Graph Dashboard

The Graph Dashboard controls what is displayed on the graph and how it is 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, air-quality, 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 <sub>2</sub> , Air-quality ...)

## 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 also be entered manually as e.g. '2018-07-18 14:00:41'. However, it is easier to use a mouse: with the mouse pointer resting within the graph area, 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.

## Scaling

Allows to change the Scaling during a run. See chapter Scaling on page 27.

## 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  $\mu\text{Sv/h}$ . If both a CPM and a CPS variable are shown at the same time, their 60fold difference may make the graph less informative. De-selecting one may be preferred.

However, when  $\mu\text{Sv/h}$  is used, the two CPM and CPS based curves will overlap!

They should also overlap if multiple counter with perhaps different tubes, or one single counter with dual tubes, as the GMC 500+ device, are used to measure a single source, but this depends on the proper use of the calibration factors for each device and tube. And this had not been the case, see discussion in chapter Set Calibrations for Geiger Tubes on page 25.

**Right-Y-axis:** This Ambient axis is used for all environmental data, like temperature, air-pressure, humidity, and air-quality. The temperature can be displayed in units of either  $^{\circ}\text{C}$  or  $^{\circ}\text{F}$ .

**Note:** Generally all data are displayed as recorded, but they can be scaled for display, while still being saved unmodified. See chapter Scaling on page 27. One example is air-pressure, which is conveniently displayed 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 remain 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 Figure 7 it shows **511.00 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  $< 10^7$ .

If the checkbox **MvAvg** is checked then a Moving Average <sup>8)</sup> 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.

The colored rectangle next to Avg (dark-blue in Figure 8) 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 32.

7 For a more detailed discussion of Normal and Poisson Distributions of Geiger data see my “[Potty Training for Your Geiger Counter](https://SourceForge.net/projects/geigerlog)” article on SourceForge <https://SourceForge.net/projects/geigerlog> .

8 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.

## 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.

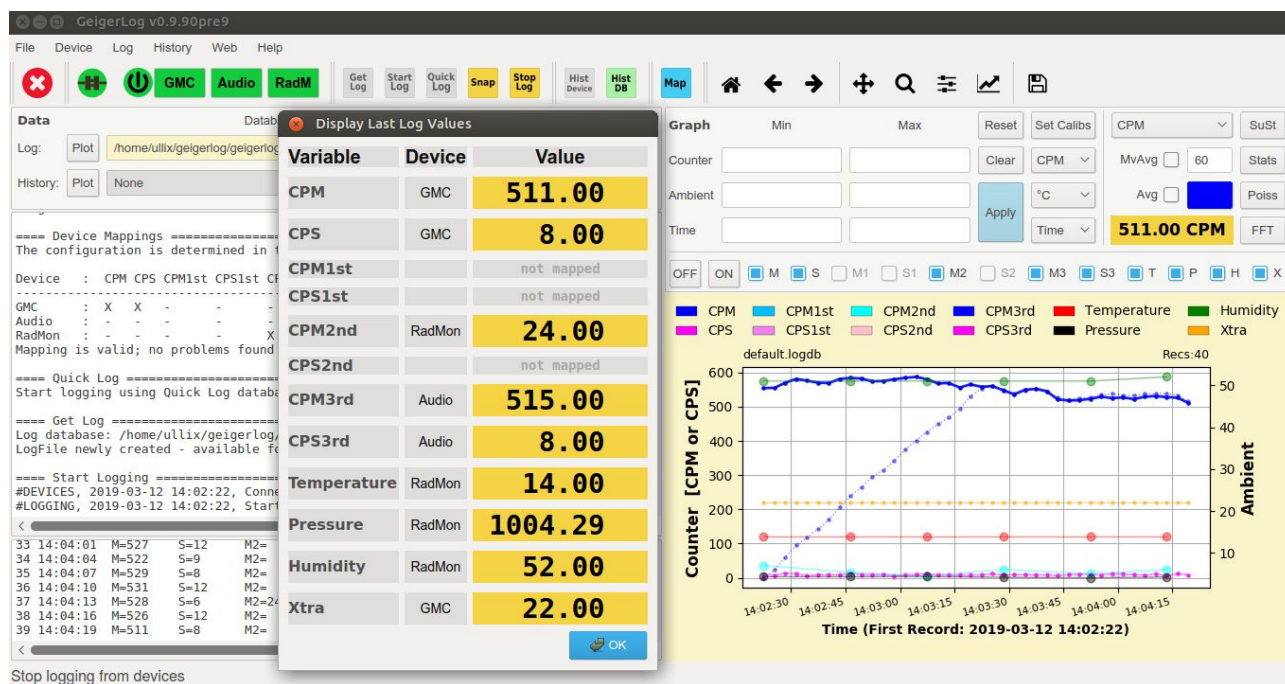


Figure 8: The Display Last Log Values Pop-up Window



# Configuration of GeigerLog

GeigerLog uses the configuration file `geigerlog.cfg` located in the folder `geigerlog`. This file is required! If it does not exist or is not readable, a STARTUP ERROR message pops up as shown in Figure 9, and GeigerLog exits.



Figure 9: Startup Error on missing or non-readable configuration file

## Complete

The options in the configuration completely determine the operation of GeigerLog. They will be read only once at the start of GeigerLog.

## Permanent versus Temporary

Most of the options are fixed for the run, but some can be changed during a run. Those changes will, however, not be written into the configuration file. So, they are temporary only, and at the next restart of GeigerLog the old options as laid out in the configuration file will again be used!

## Pure Text

The configuration file is pure text, and can be edited with any editor which does pure-text editing (see my recommendation for an editor here: Editor Geany on page 69). All options and their defaults are explained within the configuration file.

Options must be **unique**; any duplicate definitions in the configuration file result in another startup error, showing a message box like in Figure 10, giving you detailed info on the problem, and GeigerLog exits.

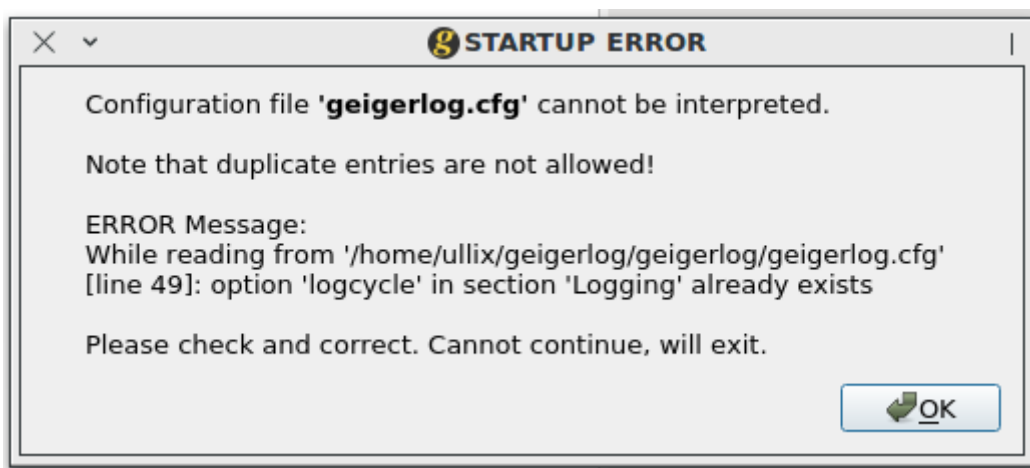


Figure 10: Startup Error on improper definitions in configuration file



## Activating devices

Devices can only be worked with if they are activated in the configuration file. If none is activated, then GeigerLog can still be used to load, show and analyze Log and History data from file.

In the default state, a GMC device and an AudioCounter device are activated, i.e. in the configuration file you will find these lines:

```
GMCActivation = yes      (in section [GMCDevice])
AudioActivation = yes    (in section [AudioCounter])
```

With the many different devices supported, some having model specific firmware bugs, changing firmware, new features and more, the configuration file may allow that even 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 90.

## Set Calibrations for Geiger Tubes

Calibration refers to the conversion of the dose rate in CPM or CPS to the dose rate in  $\mu\text{Sv/h}$  according to the formula:

$$\text{Dose Rate [CPM]} = \langle \text{Calibration Factor [CPM / } (\mu\text{Sv/h)} \rangle * \text{Dose Rate } [\mu\text{Sv/h}]$$

$$\text{Dose Rate } [\mu\text{Sv/h}] = 1 / \langle \text{Calibration Factor [CPM / } (\mu\text{Sv/h)} \rangle * \text{Dose Rate [CPM]}$$

**Important Note:** This definition of the Calibration Factor constitutes a major difference between GeigerLog Version 1.0 and the previous versions! But the difference is simple: the present use is simply the inverse of the previous use.

$$\text{new Calibration Factor} = 1 / \text{old Calibration Factor}$$

Why the change? To make things easier. Read more in chapter Appendix G – Calibration

Quick test: sort these numbers in descending order:

0.0065, 0.48, 0.002637, 0.00926, 0.09 and 0.42

How about these ones:

154, 2.08, 379, 108, 11.1, and 2.38

They are the same, except for being the inverse.

## The Dilemma

This Calibration Factor is different for each tube, and may also be different for each Geiger counter, as each device may have different shielding against the radiation it is calibrated for. An in-depth discussion of some of the issues with calibration is found in chapter Appendix G – Calibration on page 92.

Most calibration factors are provided by the manufacturer of a Geiger counter, even if only by embedding in the firmware of the device. These values are also programmed into GeigerLog and are used when the setting in the configuration file is given as:

```
calibration = auto
```

Replace 'auto' with a number to set a different calibration factor.

For the GMC Geiger counters the calibration factors are built into the firmware <sup>9)</sup> and can be read out by GeigerLog. The raw data are in the **old** units! It recently turned out that one of these factors was wrong by a factor of ~2.5!

For the tube M4011 in a GMC-300 series counter (its only tube), as well as the first tube in a GMC-500+ series double-tube counter, this factor is read out as 0.0065  $\mu\text{Sv/h/CPM}$ . For the second tube SI3BG in the GMC-500+ this number is read out as 0.194  $\mu\text{Sv/h/CPM}$ . However, that latter number is clearly wrong <sup>10)</sup>.

In GeigerLog's default configuration for a GMC-500+ the calibration factor for the 2<sup>nd</sup> tube is already set to 2.08 CPM / ( $\mu\text{Sv/h}$ ) (=0.48  $\mu\text{Sv/h/CPM}$ ), based on an experiment of my own <sup>11)</sup>.

To quickly see the current setting of calibration factors use menu **Device** → **Set Calibrations for Geiger tubes** for a dialog as in Figure 11.

Up to three different settings can be seen and edited and will be used for counter data recorded to the variables CPM1st / CPS1st, CPM2nd / CPS2nd, CPM3rd / CPS3rd, resp. For the CPM / CPS data the 1st tube factor will be used.

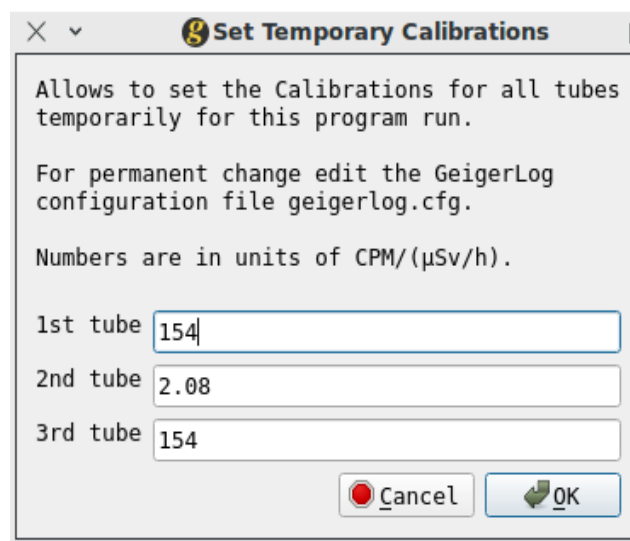


Figure 11: Set Temporary Calibrations dialog

9 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 92.

10 e.g. [http://www.ggelectronicsllc.com/forum/topic.asp?TOPIC\\_ID=5322](http://www.ggelectronicsllc.com/forum/topic.asp?TOPIC_ID=5322)

11 [http://www.ggelectronicsllc.com/forum/topic.asp?TOPIC\\_ID=5369](http://www.ggelectronicsllc.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  $\mu\text{Sv/h/CPM}$ . Which makes it 74 fold less sensitive than the M4011!”

This dialog provides a convenience function for experimentation with different tubes. It allows to redefine the calibration factor currently in use. This redefinition can be applied at any time, and its changes will be applied immediately, but it is temporary and is discarded when the program ends, or a device re-activation occurs. For lasting changes edit the configuration file `geigerlog.cfg`.

## Scaling

GeigerLog uses a powerful **Formula Interpreter** to allow scaling in support of environmental variables like temperature, air-pressure, and humidity, but this also allows dead-time correction of counter values.

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

**ValueScaling:** the measured, original 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.

ValueScaling does **not** impact data loaded from file!

**GraphScaling:** The original value is saved unmodified, and the modified value is used for plotting, and only for plotting! The air-pressure example below is a typical application and implemented by default in the configuration file.

The GraphScaling is effective also for data loaded from file.

## Formula Interpreter

Capitalization has no impact. Apart from basic math ( + - \* / ) the formula may include math functions:

+ - * /	:	basic math
**	:	raise to the power, e.g. 2**8 (=256)
VAL	:	the value to be scaled, the actual measured value
LOG	:	log to base e; natural log
LOG10	:	log to base 10
LOG2	:	log to base 2
SIN	:	sine
COS	:	cosine
TAN	:	tangent
SQRT	:	square root
CBRT	:	cube root
ABS	:	absolute value

Formula example: `val - 1000`

Formula example: `SQRT(val) * 5 + 100`

Formula example: `ABS((LOG10(val)+1000)/3.14) + 10`

The default entry is VAL, which means no modification will be applied to the measured value.

## Examples:

1. The temperature may need adjustments, which can be as simple as:

$$T_{\text{new}} = T_{\text{old}} - 0.23$$

2. Variables with a huge dynamic range – like light intensity, mapped to variable Xtra – do not allow to see small values in the plot in the presence of very large values. Square root (or cube root) allows to plot a zero value, and compress large values:

$$X_{\text{new}} = \text{Square Root } (X_{\text{old}})$$

3. Pressure data are typically within 970 ... 1030 hPa. To plot them on the same scale as other environmental parameters – like Temperature -10 ... 40 °C, Humidity 20 ... 90% – the pressure could be plotted as:

$$p_{\text{new}} = p_{\text{old}} - 1000$$

4. The air-pressure is measured at the altitude of the location of your device, but for weather stations you typically want it reduced to sea-level altitude. The formula is simple but limited to altitudes of under 50 m<sup>12</sup>).

(sea = sea-level, alt = altitude [m], Tv = “the mean annual normal value of virtual temperature at the station in kelvins”, chosen as 293 K:

$$P_{\text{sea}} = P_{\text{alt}} * (1 + \text{alt} / 29.27 * T_v)$$

5. The air-pressure is measured at the altitude of the location of your device, but for weather stations you typically want it reduced to sea-level altitude. The formula is valid at any altitude, but needs T<sup>13</sup>):

(sea = sea-level, alt = altitude [m], T = temperature [°C], T chosen as 20°C):

$$P_{\text{sea}} = P_{\text{alt}} * (1 - (0.0065 * \text{alt}) / (T + 0.0065 * \text{alt} + 273.15))^{**}(-5.257)$$

6. Dead time correction for 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 :

$$\text{CPM}_{\text{true}} = \text{CPM}_{\text{observed}} / (1 - \text{CPM}_{\text{observed}} * 200 * 1\text{E-}6 / 60)$$

$$\text{CPS}_{\text{true}} = \text{CPS}_{\text{observed}} / (1 - \text{CPS}_{\text{observed}} * 200 * 1\text{E-}6)$$

For the examples above the formulas that need to be entered in GeigerLog are:

1. @ T: VAL - 0.23
2. @ X: SQRT (VAL)
3. @ P: VAL - 1000

---

12 "Pressure Reduction Formula", WORLD METEOROLOGICAL ORGANIZATION, CIMO/ET-Stand-1/Doc. 10, (20.November.2012), chapter 3.11.2 Low-level stations

Quote: “At low-level stations (namely, those at a height of less than 50 m above mean sea level), pressure readings should be reduced to mean sea level by adding to the station pressure a reduction constant C given by the following expression:  $C = p \cdot H_p / 29.27 T_v$  (3.3).

$H_p$  is the station elevation in meters; and  $T_v$  is the mean annual normal value of virtual temperature at the station in kelvins.”

13 <https://keisan.casio.com/exec/system/1224575267>

4. for an altitude of 49 m and  $T_v = 293$  K  
@ P:  $VAL + VAL * 49 / (29.27 * 293)$
5. for an altitude of 85 m and a temperature of 20 °C:  
@ P:  $VAL * (1 - 0.0065 * 85) / (20 + 0.0065 * 85 + 273.15) ** (-5.257)$
6. for a dead time of 200µsec (use at any CPM\* or CPS\*):  
@ CPM:  $int(VAL / (1 - VAL * 200 * 1E-6 / 60))$   
@ CPS:  $int(VAL / (1 - VAL * 200 * 1E-6))$   
NOTE: use the int() to convert floating point numbers to integer numbers in order to not confuse the Poisson test with fractional numbers in CPM or CPS!

## View and Edit Current Scaling Dialog

To view the current setting of your scaling, click menu **Help** → **View and Edit Current Scaling Dialog** or click **Scaling** button in the **Graph Dashboard** to see a dialog as in Figure 12.

You can edit all entries. They become active after you click OK, but remain in effect **only for the current run**. They are NOT saved to the configuration file!

See GeigerLog manual in chapter **Configuration of GeigerLog**, topic **Scaling**, for guidance

ValueScaling - it DOES modify the saved value!		GraphScaling - it does NOT modify the saved value, only the plotted value	
CPM	VAL	CPM	VAL
CPS	VAL	CPS	VAL
CPM1st	VAL	CPM1st	VAL
CPS1st	VAL	CPS1st	VAL
CPM2nd	VAL	CPM2nd	VAL
CPS2nd	VAL	CPS2nd	VAL
CPM3rd	VAL	CPM3rd	VAL
CPS3rd	VAL	CPS3rd	VAL
T	VAL	T	VAL
P	VAL	P	VAL - 1000
H	VAL	H	VAL
X	VAL	X	VAL

Figure 12: View and Edit Current Scaling Dialog

## WorldMap

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

To obtain this information you need to register at <http://www.gmcmap.com/> and obtain a UserID (probably 5 digits) and a CounterID (probably 11 digits). User and Counter IDs valid for the gm-cmap.

# 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**<sup>14</sup>). You will be offered to load an existing \*.log, \*.his, \*.CSV, or \*.txt file, which will then be presented in the dialogue. An example is in Figure 13.

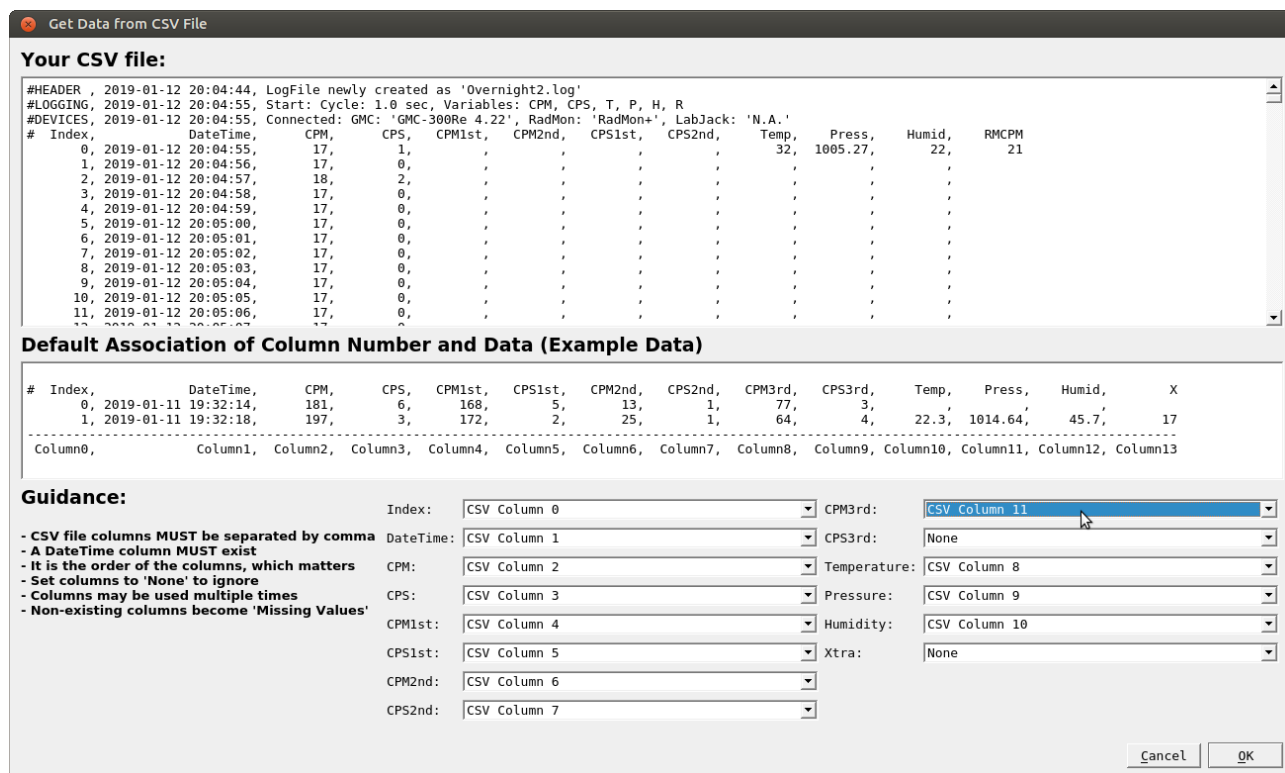


Figure 13: The Get Data from CSV File dialogue

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 CSV formatted \*.log, \*.his Files

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.

<sup>14</sup> The two actually aren’t different; it just helps to organize any data by their way of creation

# 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 menu options **File** → **Show Scatter Plot from Plot Data** and **File** → **Show Plot Data** 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. The SuSt function uses all the variables in the plot. 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 20.

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  $\mu\text{Sv/h}$ . Temperature may be shown in  $^{\circ}\text{C}$  or  $^{\circ}\text{F}$ .

## 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](https://SourceForge.net/projects/geigerlog/)” 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  $\mu\text{Sv/h}$  is used!

Likewise, for the variables temperature, air-pressure, humidity, and air-quality 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 logcycle: 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,      RMCPPM
    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  $\mu\text{Sv/h}$  (or any other dose rate, like  $\text{mR/h}$  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; Figure 14 for low count rates as in a background measurement, and Figure 15 for a much higher count rate.

The value  $r^2$  (in the graph as  $r2$ ) is an indicator for the goodness of a fit. **A value of  $r^2 \geq 0.9$  suggests a proper measurement.** If  $r^2$  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.

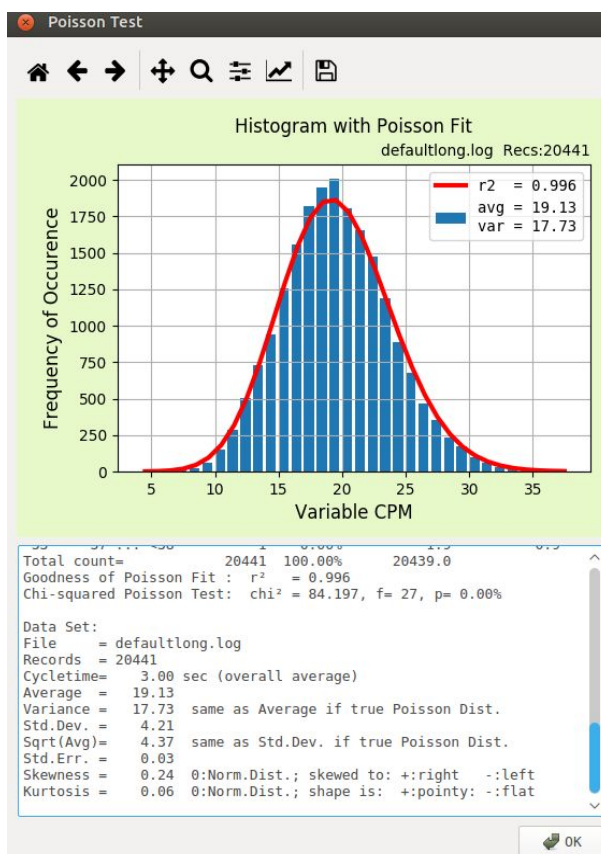


Figure 14: Histogram of Low Count Rate

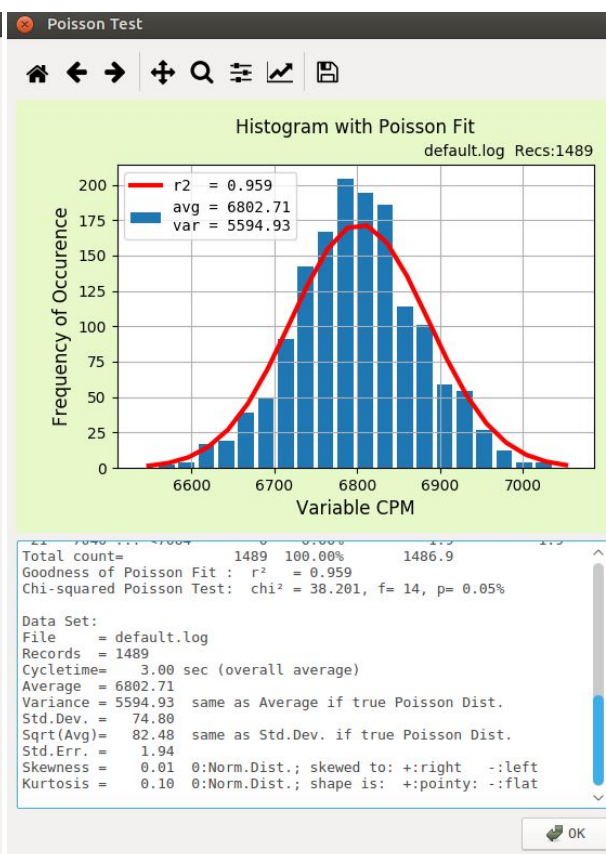


Figure 15: Histogram of High Count Rate

## 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 Figure 16. The data were recorded by logging in the CPM mode.

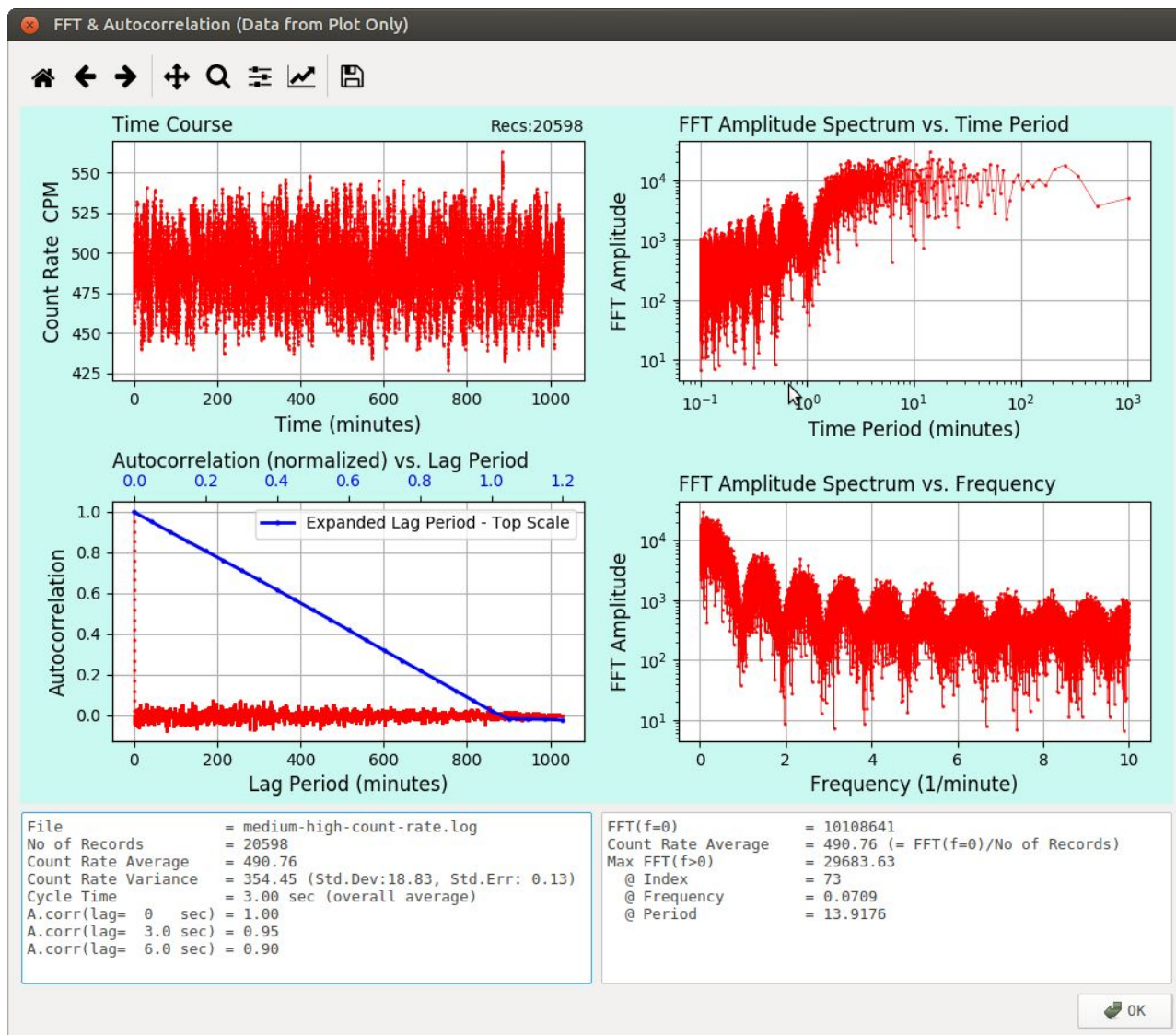


Figure 16: FFT Analysis of Medium Count Rate Measurement

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 ( $< 30$ sec). At a sampling time of  $> 30$ sec, 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 Figure 16. 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 Poisson distribution of the Geiger data. The FFT spectrum is then a mix of the rectangle spectrum and the Poisson spectrum.

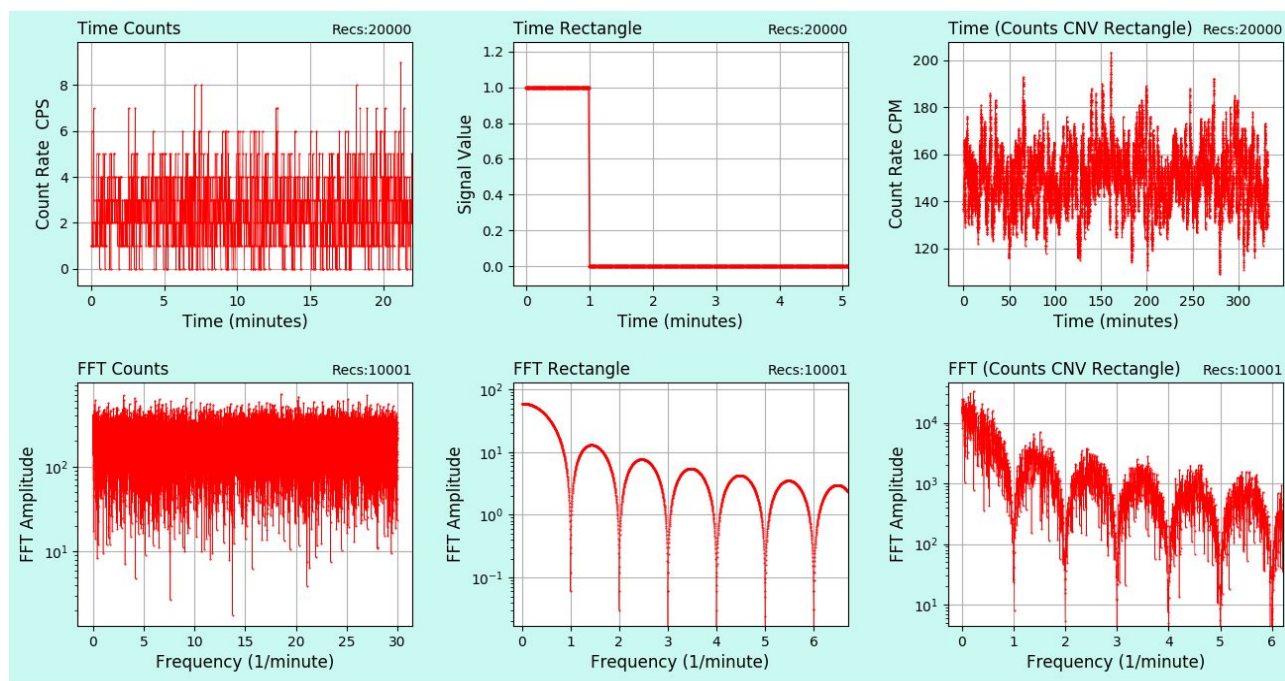


Figure 17: Demonstration of the impact of convolution on an FFT spectrum

(Synthetic data, CPS=2.5)

This can be nicely demonstrated using synthetic recording. In Figure 17 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.



## 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?

More relevant to Geiger counters was the recent observation <sup>15)</sup> that a GMC Geiger counter equipped with a M4011 tube (black version) was significantly sensitive to temperature, which it shouldn't be!

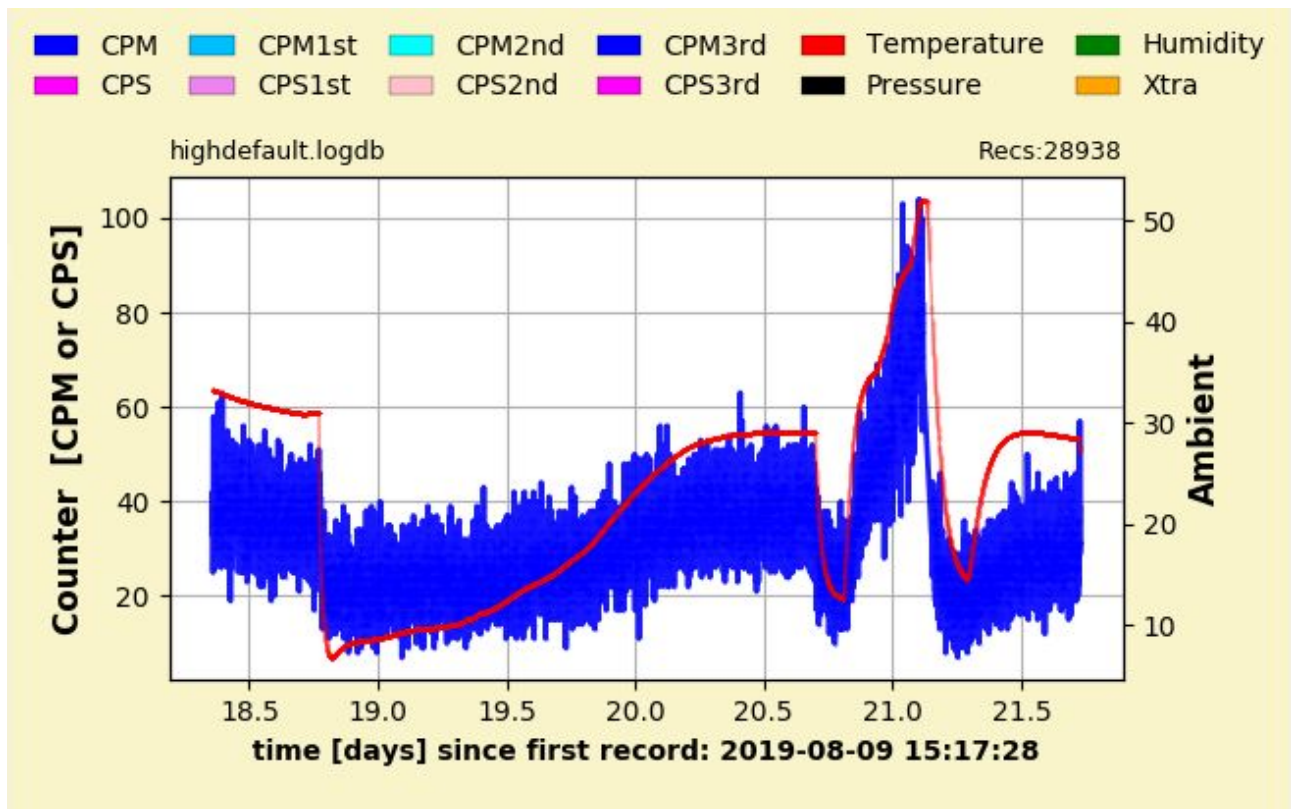


Figure 18: CPM and Temperature in a Temperature Experiment with a Geiger Counter (GMC-300E+ with a M4011 tube - black version)

It is obvious from the time course plot in Figure 18 that CPM (blue curve) increases and decreases with temperature (red curve). The degree of this correlation can be better demonstrated by plotting a scatter plot of CPM versus temperature.

Use the command in menu **File** → **Show Scatter Plot from Plot Data**. In the upcoming dialog you choose the variables for the X- and Y-axis, which in this example will be X = temperature and Y = CPM as shown in Figure 19.

You can also choose whether you want to show X, or Y, or both to include their origin (0, zero) in the plot, and whether to draw connecting lines between data points. Default is no showing of zero, and drawing connecting lines, which on most occasions is the best option.

Furthermore, you have the option of including a polynomial Least-Squares-Regression fit to the data, and can choose the order of the polynomial from “Prop” ( $y=m*x$ ), 0 (equals average) to 7.

<sup>15</sup> [http://www.ggelectronicssl.com/forum/topic.asp?TOPIC\\_ID=7475](http://www.ggelectronicssl.com/forum/topic.asp?TOPIC_ID=7475)

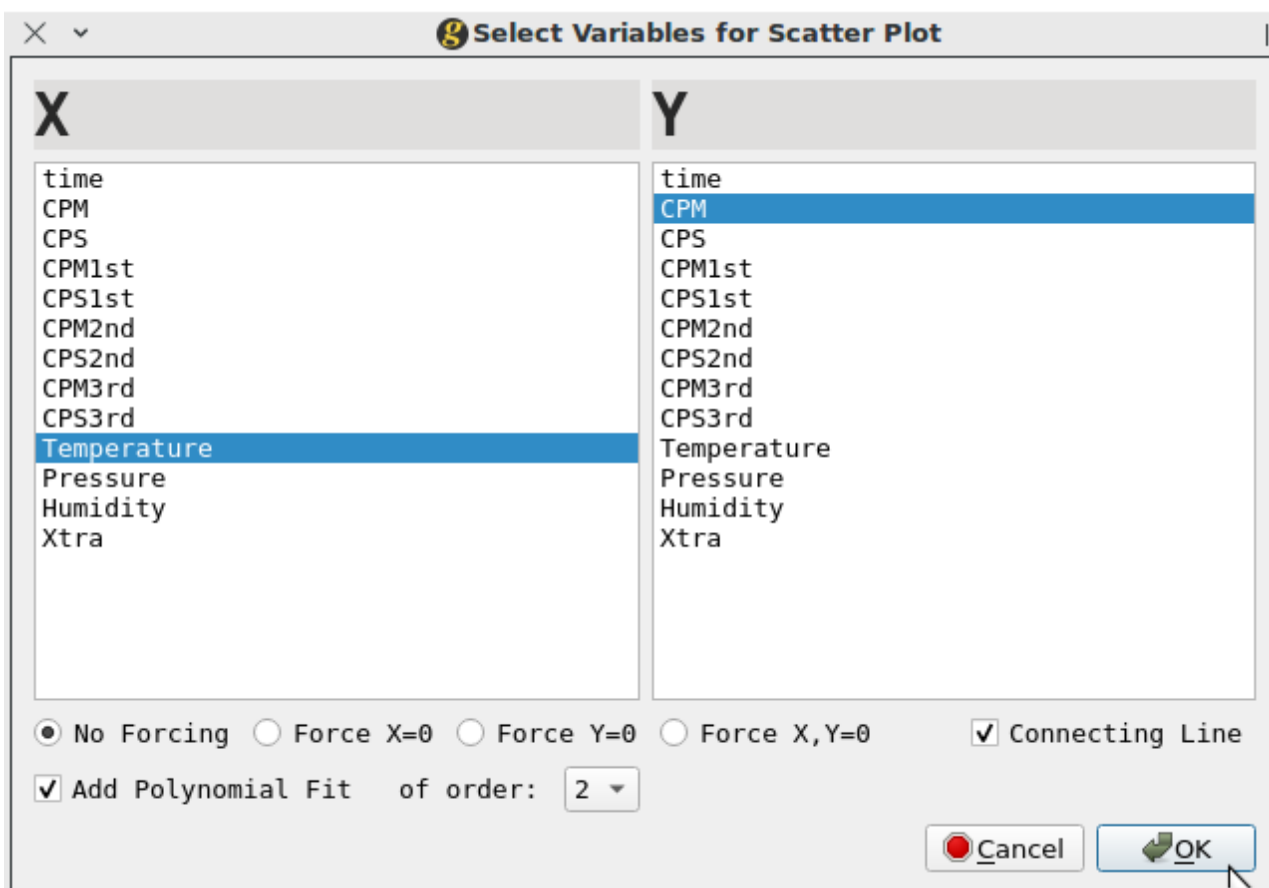


Figure 19: Select Variables for Scatter Plot

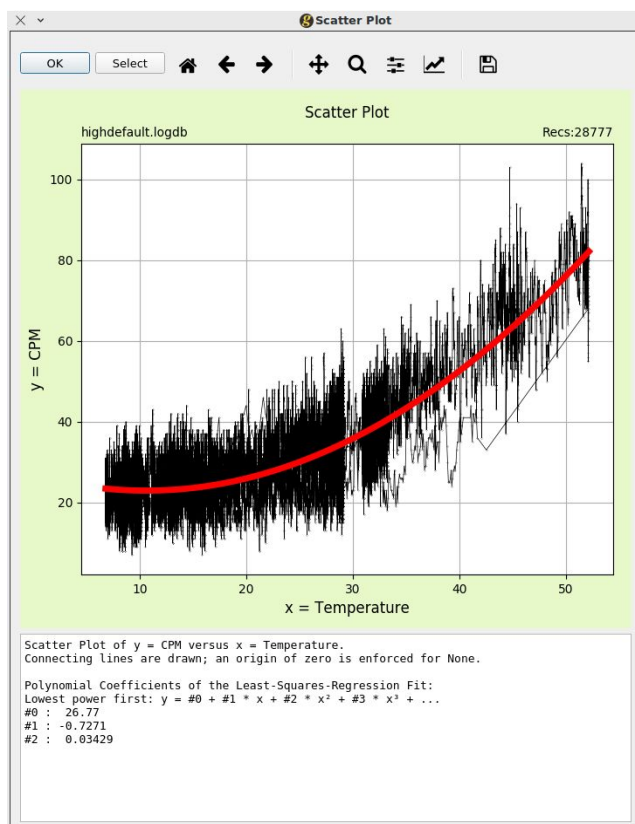


Figure 20: Scatter Plot with 2nd Order Polynomial Fit

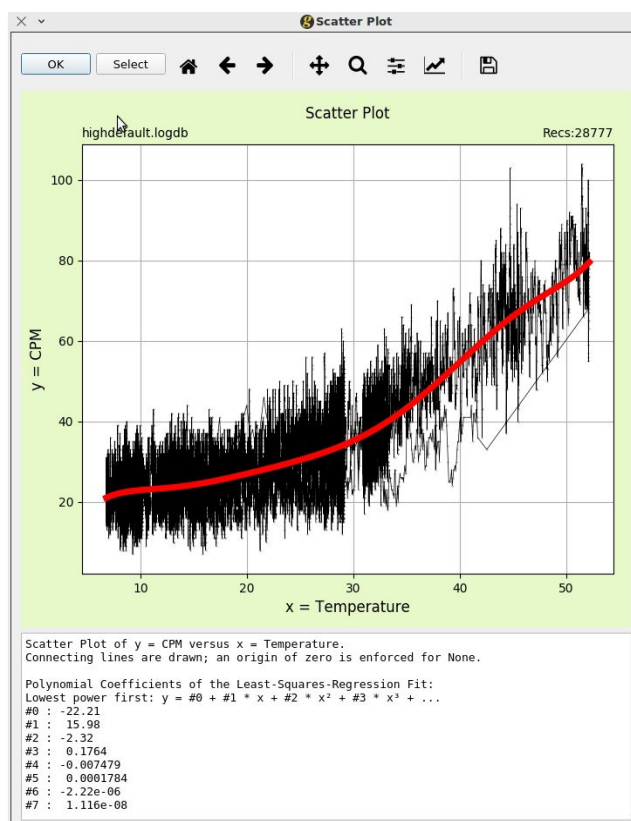


Figure 21: Scatter Plot with 7th Order Polynomial Fit

In Figure 20 you see a scatter plot of  $y = \text{CPM}$  versus  $x = \text{Temperature}$  plot in black, fitted with a polynomial of second order (also called a quadratic fit) in red.

Clicking the OK button (top left) will close the scatter plot. The Select button brings you back to the variable selection as shown in Figure 19. The choices you had just made, are maintained. Let's change the order of the fit from 2 to 7, and click OK.

Figure 21 shows the result with now a fit of the same data with a polynomial of 7<sup>th</sup> order.

With these data one can barely see a difference between the two fits. However, with other data the high order fits may show curvature which simply is meaningless! Caution must be applied to choosing an order for the fit; high orders may result in over-interpretation of data. Generally a lower order is more adequate; more than a 2<sup>nd</sup> order will rarely ever be needed nor meaningful!

Another issue which demonstrates the usefulness of a Scatter Plot 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 46.

In the Select Variables dialog (Figure 19) choose the variables as  $X = \text{CPM}$  (digital data) and  $Y = \text{CPM3rd}$  (audio data). Uncheck **Connecting Line**, and choose a fit of order = 1, i.e. a linear fit. The result is shown in Figure 22, demonstrating the excellent correlation between the two.

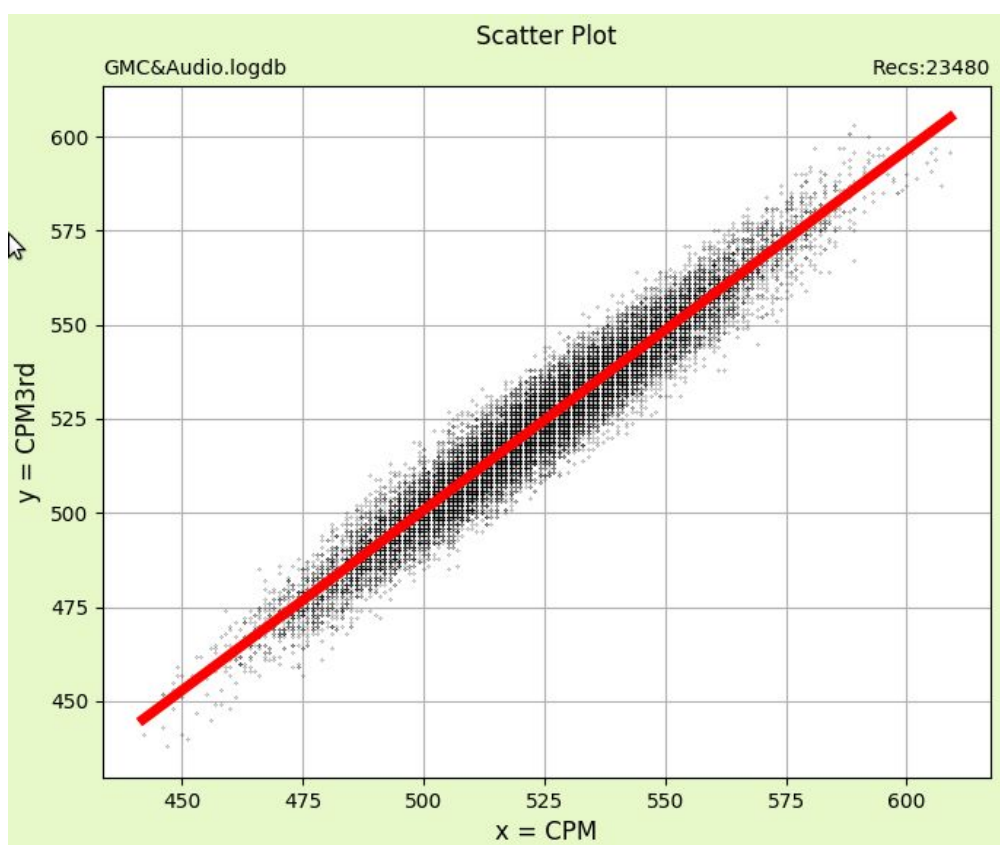


Figure 22: Scatter Plot of CPM3rd=Audio Signal versus CPM=Digital Signal (Using a GMC-300E+ Geiger Counter)

## Printing Values of the Variables

Sometimes you want the numerical values of your variables. Use the command in menu **File** → **Show Plot Data** 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, as shown in Figure 23. 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 with **File** → **Print NotePad**.

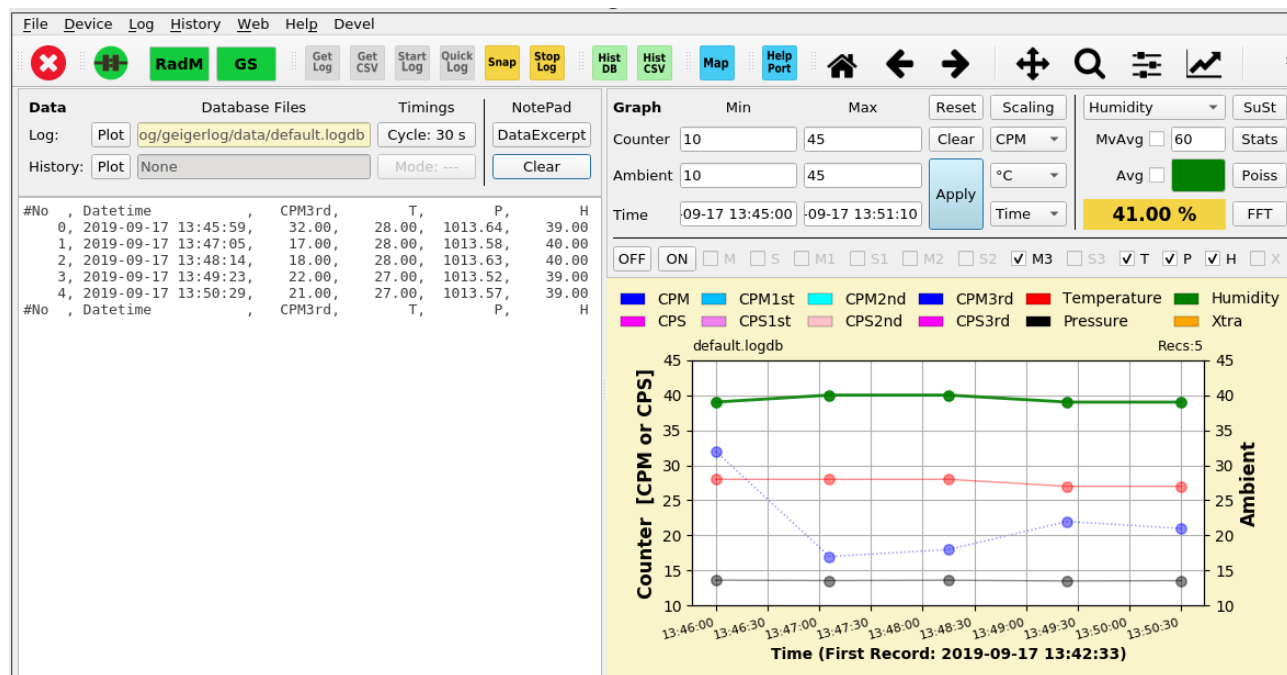


Figure 23: Command File --> Show Plot Data showing only the data currently in the plot



# 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

First see chapter Appendix B – Connecting Device and Computer using a Serial Connection on page 75 for an important explanation!

After starting GeigerLog and with proper USB Port and baudrate established, select menu **Device** → **Connect Device** as explained in Establishing a GeigerLog Connection to Your Devices on page 14.

If unsuccessful, a printout **in red** will tell you the reason. You will likely be advised to run **Autodiscover USB Port for Device** → **GMC** from menu **Help**.

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 90.

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 <sup>16)</sup>!

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 <sup>17)</sup>)  
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

---

<sup>16</sup> 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.

<sup>17</sup> See discussion e.g. here: [http://www.ggelectronicsllc.com/forum/topic.asp?TOPIC\\_ID=5304](http://www.ggelectronicsllc.com/forum/topic.asp?TOPIC_ID=5304)

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:

$$\begin{aligned}\text{CPM} &= \text{CPM1st} = \text{CPM2nd} \\ \text{CPS} &= \text{CPS1st} = \text{CPS2nd}\end{aligned}$$

An example of a Logging with a GMC-device in combination with an AudioCounter as a 2<sup>nd</sup> device is shown in Figure 24 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.

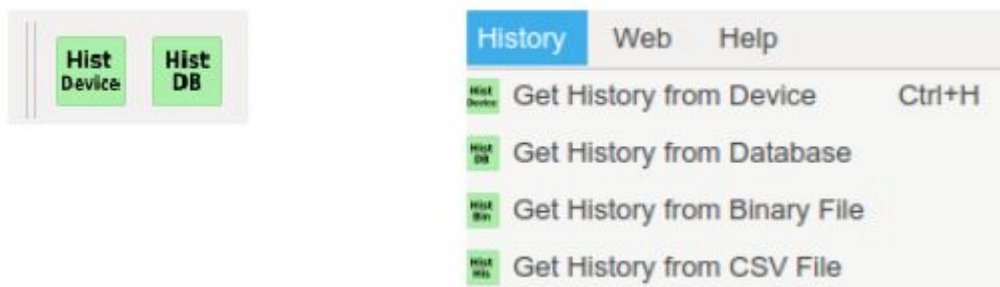
In theory you could download only a portion of the memory. But since this is laid out as a ring-buffer, you'd have to know very precisely what portion of the memory you want. Typically you won't know this until after you have done the complete download and inspected the data. On top of this, a partial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

**NOTE:** if you experience reading errors while downloading the history, or even 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 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.

**History** → **GMC Series** → **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.

A History example – data collected during a long distance flight – is shown in Figure 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 of the Device

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. Such counters are now fully support by GeigerLog. How to operate, connect, configure and run such devices is described in the recently published article **GeigerLog-AudioCounter-Support**<sup>18)</sup>.

An example for a low-cost, audio-only counter is the recently reviewed **Radiation-v1.1(CAJOE)** counter<sup>19)</sup>.

But even some very advanced counter designs offer only an audio output, like in particular this also recently reviewed, semiconductor based radioactivity detector **Smart Geiger Pro (SGP-001)**<sup>20)</sup>.

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. Figure 24 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.

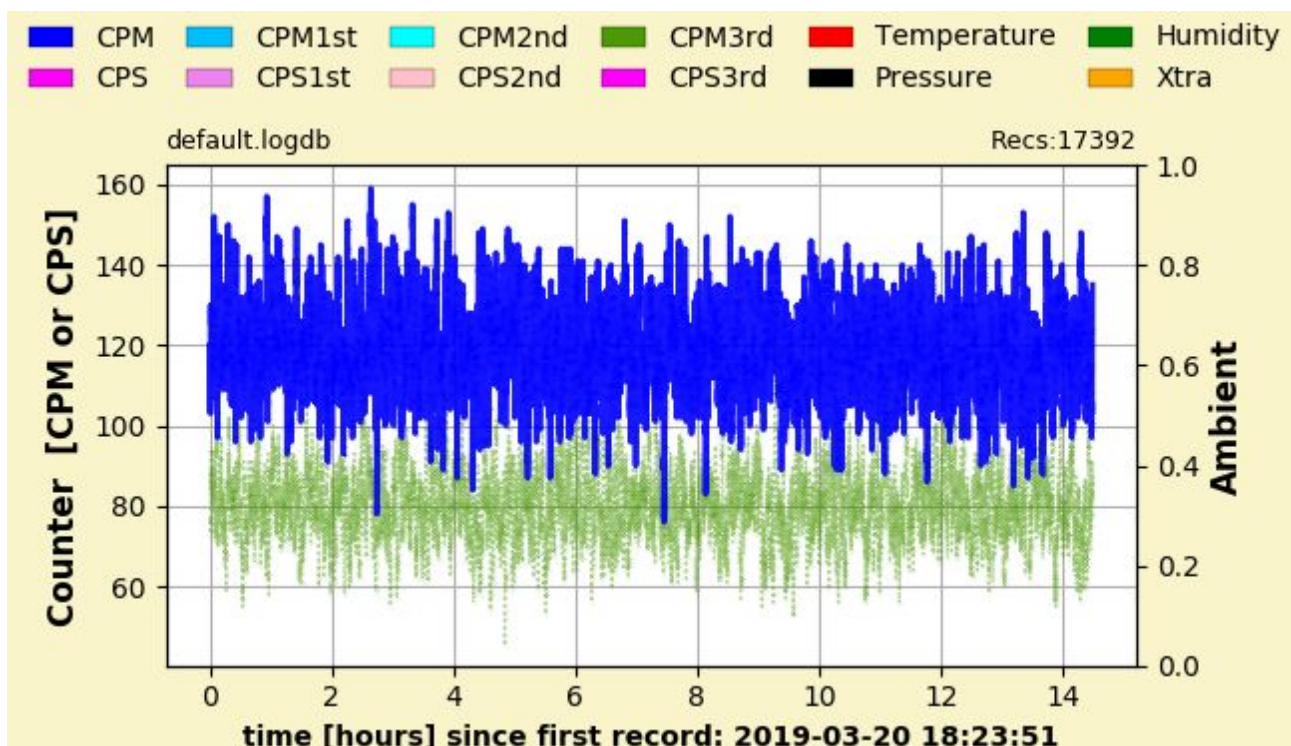


Figure 24: GMC-300E+ (blue) digital recording, and SGP-001 Device (green) audio recording

18 Read article **GeigerLog-AudioCounter-Support-v2.0.pdf** on Sourceforge at:  
<https://sourceforge.net/projects/geigerlog/files/Articles/GeigerLog-AudioCounter-Support-v2.0.pdf/download>

19 Read article **GeigerLog-Radiation-v1.1(CAJOE)-Support** on Souceforge at  
<https://sourceforge.net/projects/geigerlog/files/Articles/GeigerLog-Radiation-v1.1%28CAJOE%29-Support-v1.0.pdf/download>

20 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/>, <sup>21</sup>).

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)** <sup>22</sup> on Sourceforge.

### Connecting

First see chapter Appendix B – Connecting Device and Computer using a Serial Connection on page 75 for an important explanation! Since the ELV device identifies itself with its name – see Figure 35 – it is easy to select the proper port with a baudrate of 115200 from menu **Help** → **Show & Select USB Port and Baudrate**.

With proper USB Port and baudrate established, select menu **Device** → **Connect Device** as explained in Establishing a GeigerLog Connection to Your Devices on page 14.

Figure 25 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.

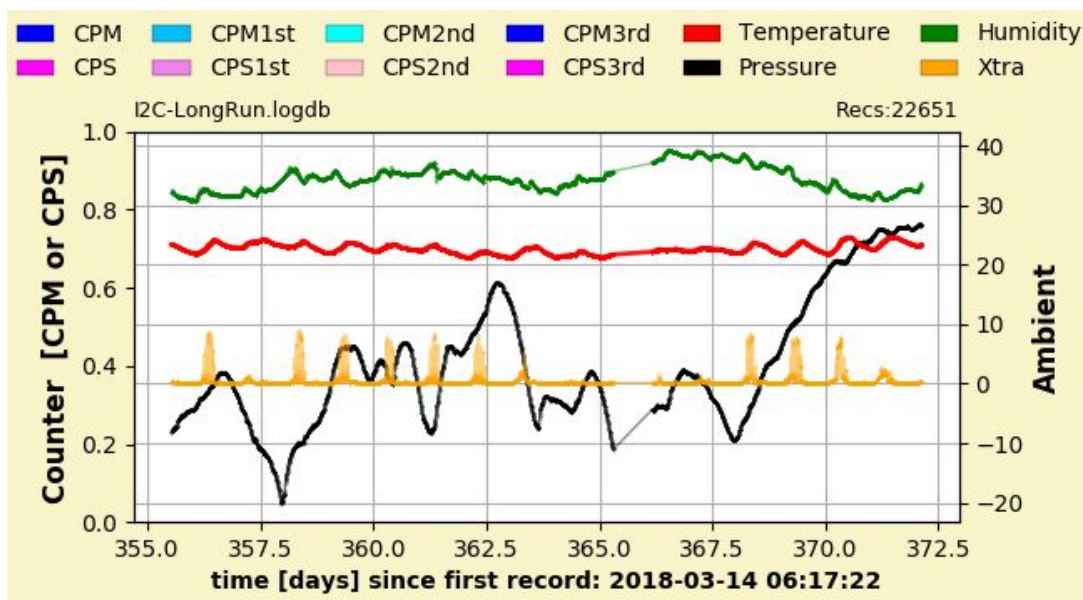


Figure 25: A Long-Term recording Using I2C-Devices

<sup>21</sup> The most recent version is <https://sourceforge.net/projects/i2cpytools/files/I2Cpytools-v0.2.zip/download>

<sup>22</sup> <https://sourceforge.net/projects/i2cpytools/files/I2Cpytools-Manual-v0.2.pdf/download>

# RadMon Devices

The RadMon devices <sup>23)</sup> 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 <sup>24)</sup> 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 <sup>25)</sup> 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 <sup>26)</sup> – 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 <sup>27)</sup>. The total of reading the 3 environmental

---

23 DIYGeigerCounter <https://sites.google.com/site/diygeigercounter/>

24 [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things)

25 BOSCH BME280

26 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.



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 <sup>28</sup>). 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 of the Device

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 definitely want to have the manual <sup>29</sup>) ready when you do the assembly!

## 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.

You have to bring the RadMon+ into its configuration mode, which confuses a little bit. From the manual I have copied the starting sequence, and inserted here as Figure 26. For the rest you can follow the instructions on the smartphone. When done, press the RST button on the RadMon+.

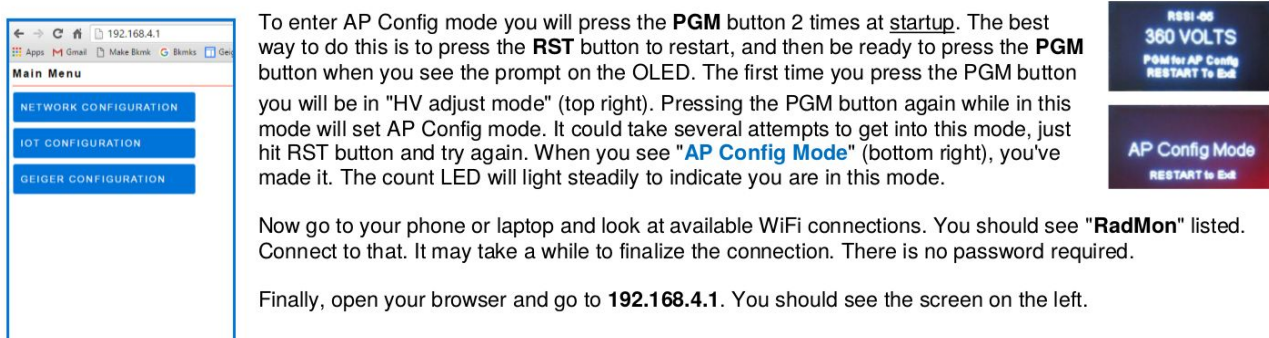


Figure 26: Activating RadMon Configuration on a Smartphone

27 Initially I used the server [iot.eclipse.org](https://iot.eclipse.org), but it seems to have been switched off. More recently I used: [broker.hivemq.com](https://broker.hivemq.com). However, many more such servers are publicly available for testing purposes: [https://github.com/mqtt/mqtt.github.io/wiki/public\\_brokers](https://github.com/mqtt/mqtt.github.io/wiki/public_brokers)

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

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

192.168.4.1/general.html
Geiger Configuration

192.168.4.1/iot.html
IoT Configuration

Count Period: 60  
Sleep Period: 0  
CPM to uSv: 175.00  
Elevation (m): 0  


---

Use BME280? ☒  
Adafruit BME280? ☐  
Use millibar? ☒  
Use Celsius? ☒  
xmit Charge Status? ☐  
DEBUG on? ☐  
SAVE

Radmon.org?  
User RADMON USER NAME  
Password RADMON PASSWORD  


---

MQTT? ☒  
User  
Password  
Server iot.eclipse.org  
Port 1883  
Topic Path geigerlog/  


---

ThingSpeak?  
writeAPI THINGSPEAK WRITE API  
SAVE

Figure 27: RadMon+ Configuration pages

The first page needs info on your own wireless LAN, so that RadMon+ can connect to it. The next two pages are shown in Figure 27 as screen shots from my smart-phone:

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 ‘geiger-log/’ – ending with a slash ‘/’! The same must be entered into Geiger-Log’s configuration file (see below) as ‘RMServerFolder = geigerlog/’!

Figure 28 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 weather-proof housing. The CPM average with a SBM20 tube is 20.1 CPM.

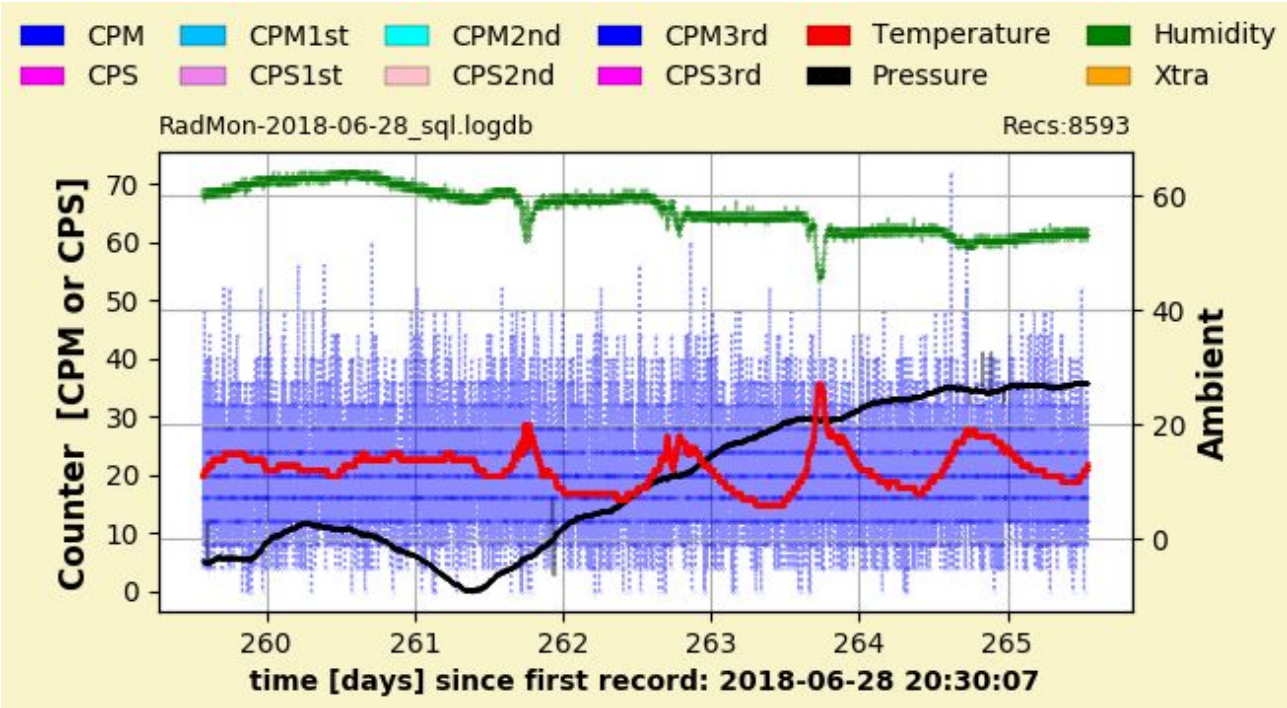


Figure 28: Long-Term Recording from an Outside RadMon+ Weather Station

# 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`

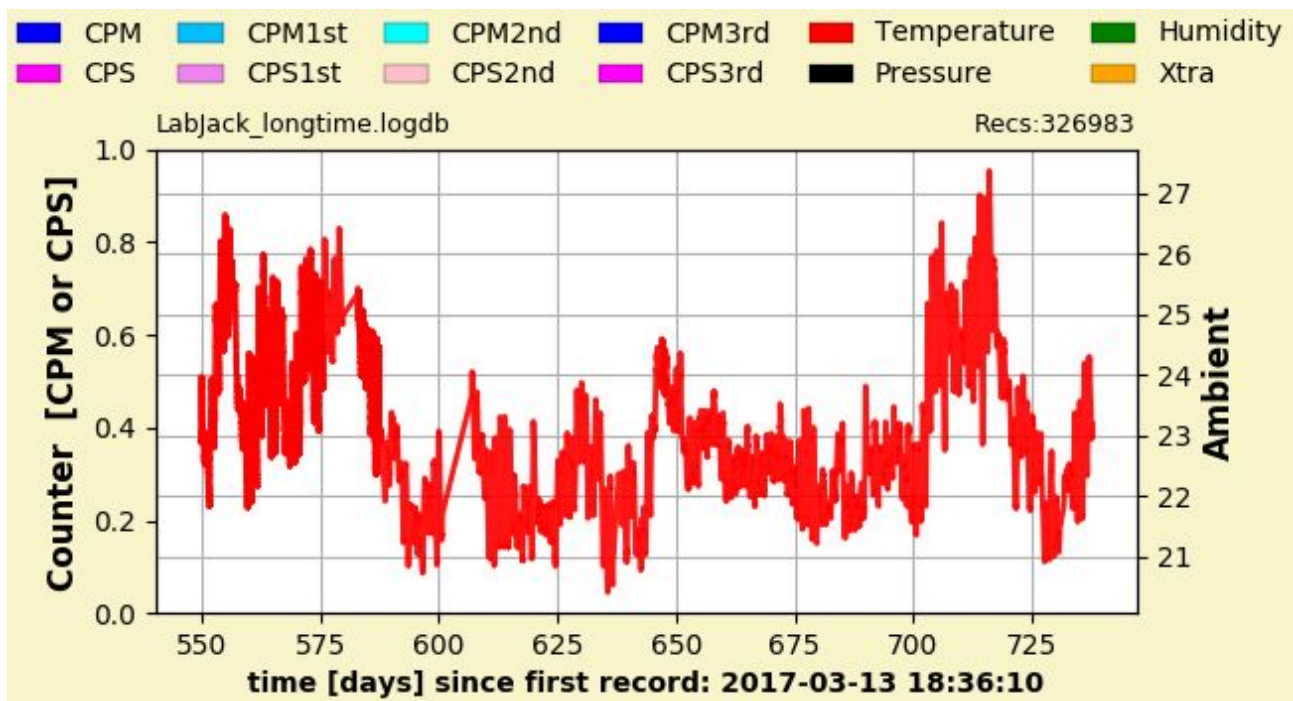


Figure 29: Long-Term Recording – 2 years – with the LabJack, showing Temperature only

# 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 4<sup>th</sup> 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

See chapter Appendix B – Connecting Device and Computer using a Serial Connection on page 75 for a detailed explanation!

Since the Gamma-Scout device identifies itself with its name (like: Gamma-Scout USB Serial Port Driver (COM5)) it is easy to select the proper port with a baudrate specific for this device (probably 9600) from menu **Help** → **Show & Select USB Port and Baudrate**.

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 GeigerLog configuration file `geigerlog.cfg`!

## 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!

Next lines are an excerpt of a History download from a Gamma-Scout device:

```
#>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
```



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 line 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 at 30 and 36, resp.

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

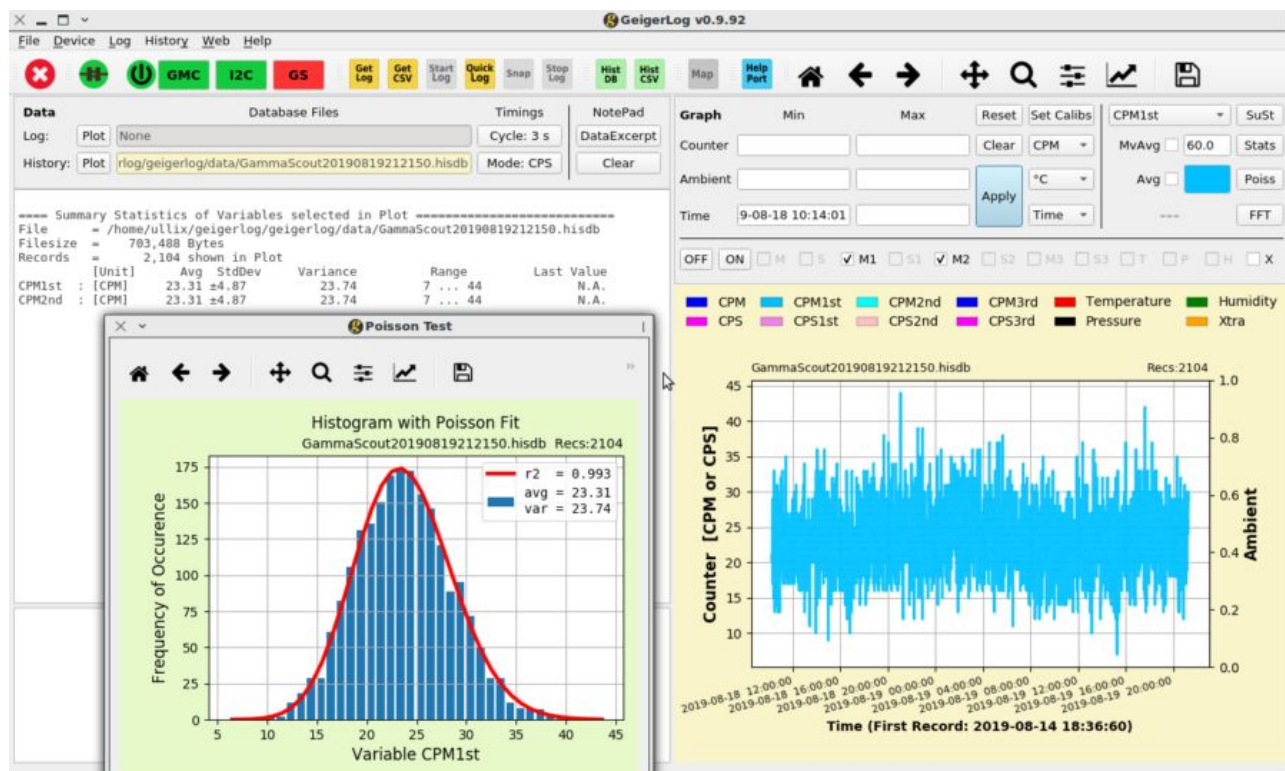


Figure 30: A recording from a Gamma-Scout device

The CPM data from variable CPM1st do form a proper Poisson test.

# 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 Full Log Re-Plot the full log file data (if loaded)
- Plot Full History Re-Plot the full history file data (if loaded)
- Show Scatterplot from Plot Data Plot one variable against another in a x-y-scatter plot, using only the data currently shown in the plot
- Show Plot Data Show the Date&Time and variable values as currently shown in the plot
- Show Plot Data Summary Statistics (SuSt) Print summary statistics for all data currently shown in the plot to the NotePad
- Show Plot Data Statistics In a pop-up window show detailed statistics for the data currently shown and selected in the plot
- Show Plot Data Poisson Test In a pop-up window show a ‘Histogram with Poisson Fit’ for the data currently shown and selected in the plot
- Show Plot Data 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

- Show Device Mappings                      Show the mapping of variables to the activated devices
- Set Calibrations for Geiger Tubes Set calibrations for all Geiger tubes temporarily

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
- 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 be created.
- Add Comment to Log Adds a comment to the log file; does not disturb logging or graphing.

- 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
- 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
- Submenu: AmbioMon Series           History functions related to AmbioMon 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 Geiger-Log.

## 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 – History – Submenu: AmbioMon Series

- Get History Binary CAM Data 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!

A database file with the extension \*.hisdb will be created.

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 Binary CPS Data 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!

A database file with the extension \*.hisdb will be created.

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 Binary CAM Data from File

Getting Data from File instead from Device

- Get History Binary CPS Data from File

Getting Data from File instead from Device

## Menu – Web

- Update Radiation World Maps      Upload your current data to the Radiation World Maps, see Radiation World Maps on page 71.

## 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 and Edit 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.
- **Submenu: Autodiscover USB Port for Device:**      GeigerLog makes an attempt to determine port and baud rate for a GMC Geiger counter, or an I2C device, or a Gamma-Scout Geiger counter automatically, and shows the report in a pop-up dialog box.  
  
But be aware of side effects, see Appendix B – Connecting Device and Computer using a Serial Connection on page 75 for more details.
- **About GeigerLog**      A brief introduction to GeigerLog, as well as version and legal information

### **Menu – Help – Submenu: Autodiscover USB Port for:**

- **GMC**      Autodiscover the USB Port for any GMC device.
- **I2C**      Autodiscover the USB Port for any I2C device.
- **GS**      Autodiscover the USB Port for any Gamma-Scout device.

# Toolbars

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

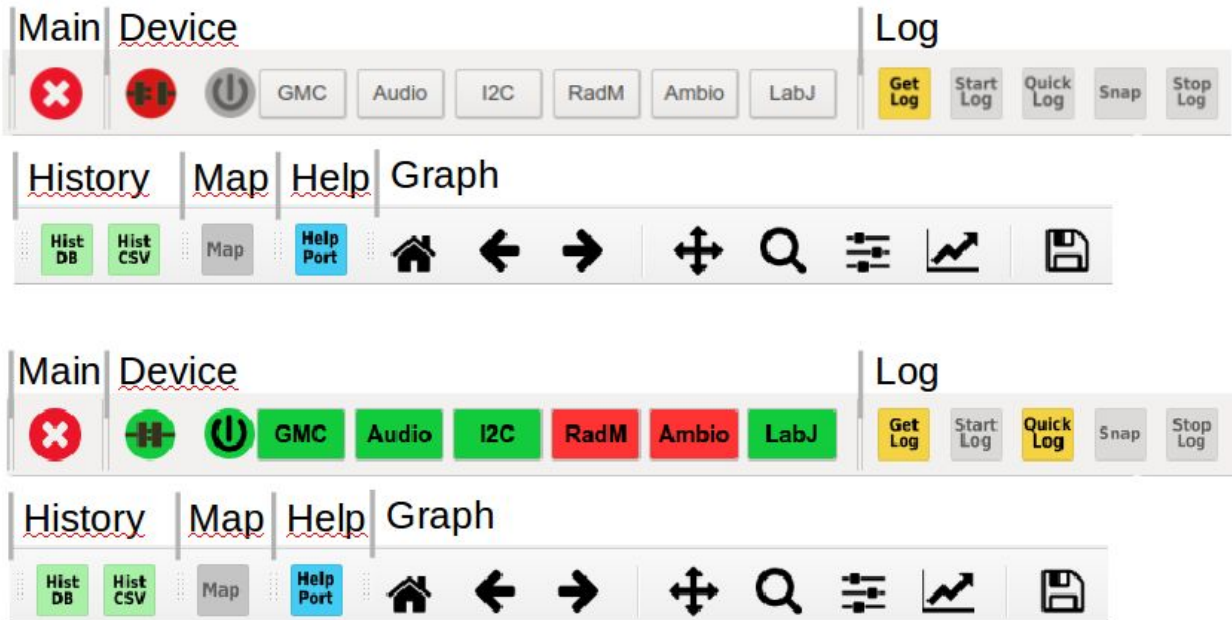


Figure 31: The toolbars

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.

The meaning of the icons in the toolbars:

- 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: successfully connected, otherwise color= red. The figure shows:
    - 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)
- Not shown: a Gamma-Scout device, as it had not been activated!
  
- Log
  - Get a Log File
  - Start Logging
  - Start a Quick Log
  - Snap a single record during logging
  - Stop Logging
  
- History
  - Get a History file from database
  - Get a History file from CSV file
  
- Map
  - Upload your current data to the Radiation World Maps, see Radiation World Maps on page 71.
  - 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.

# Helpful Internal Software Tools

These internal tools come as part of GeigerLog and can be found in the GeigerLog directory. None is required to run GeigerLog, but sometimes can be really helpful.

## GLpipcheck

It tells you which packages required by GeigerLog are installed or are missing, and which ones have upgrades available. Very helpful during the installation. Start with:

```
python3 /path/to/GLpipcheck
```

for output like this:

```
----- GLpipcheck -----
Python Executable: /home/ullix/geigerlog/vgl38/bin/python3
Python Version:    3.8.6 (default, Oct 6 2020, 04:02:53) [GCC
5.4.0 20160609]
```

Listing of all Pip-found packages:

Package	Version
certifi	2020.6.20
cffi	1.14.2
kiwisolver	1.2.0
matplotlib	3.3.2
numpy	1.19.1
paho-mqtt	1.5.1
Pillow	7.2.0
pip	20.2.3
pip-check	2.6
pycparser	2.20
pyparsing	2.4.7
PyQt5	5.15.1
PyQt5-sip	12.8.1
pyserial	3.4
python-dateutil	2.8.1
scipy	1.5.2
setuptools	50.3.0
six	1.15.0
sounddevice	0.4.1
SoundFile	0.10.3.post1

GeigerLog required packages and their currently installed versions:

	Package	Version
installed:	matplotlib	3.3.2
installed:	numpy	1.19.1
installed:	paho-mqtt	1.5.1
installed:	pip	20.2.3
installed:	pip-check	2.6
installed:	PyQt5	5.15.1

```
installed: PyQt5-sip          12.8.1
installed: pyserial           3.4
installed: scipy              1.5.2
installed: setuptools         50.3.0
installed: sounddevice        0.4.1
installed: SoundFile          0.10.3.post1
```

GeigerLog required packages missing:  
None

GeigerLog required packages having Upgrades available:

	Package	Version	Latest	Type
installed:	numpy	1.19.1	1.19.2	wheel

## GLsoundcheck

GLsoundcheck allows to check the audio signal for use as an AudiCounter input by plotting a live microphone signal. Very helpful for inspecting the audio input and setting the audio amplification.

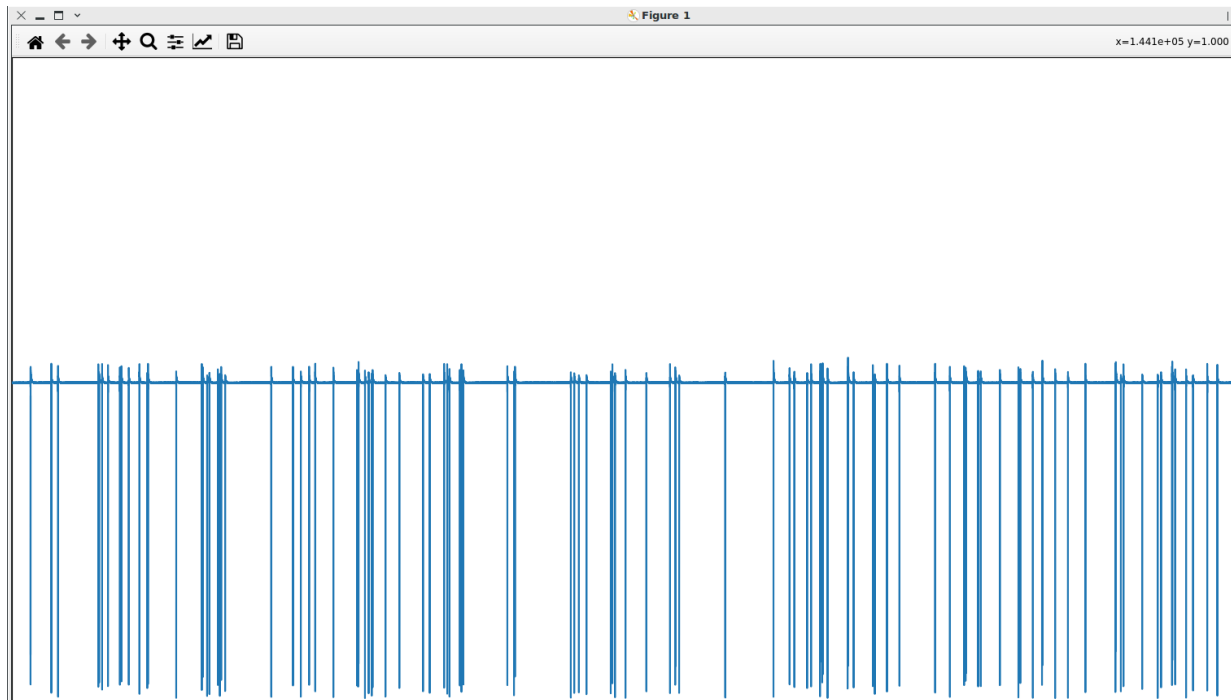
Start with:

```
python3 /path/to/Glsoundcheck
```

Stop with:

```
CTRL-C
```

Its graphic output looks may look like this (a 20 sec recording sampled with 44100 samples per second for negative audio signal):



*Figure 32: Output from the Glsoundcheck for an audio source with negative pulses*

## GLsermon

GLsermon allows to monitor the serial port and to send commands via the USB-to-Serial Connection. Very convenient to monitor a counter's response to commands issued via serial.

Start with:

```
python3 /path/to/GLsermon PORTNAME BAUDRATE
PORTNAME on Linux    = /dev/ttyUSBx, x = 0, 1, 2, ...
           on Windows = COMx, x = 0, 1, 2, ...
BAUDRATE              = 57600, 115200, ...
Starting with 'GLsermon' defaults to
PORTNAME = /dev/ttyUSB0, BAUDRATE = 57600
```

Stop with:

CTRL-C

A communication session with GLsermon connected to a GMC-300E+ Geiger counter – both input and output will be recorded to file `Glsermon.txt` – may look like this:

```
Command : <GETVER>>
Response: GMC-300Re 4.22
R-Values:  71  77  67  45  51  48  48  82 101  32  52  46  50  50

Command : <GETVOLT>>
Response: *
R-Values:  42

Command : <GETCPS>>
Response: (Non-decodeable)
R-Values: 128  7
```

Command is what is sent to the counter, Response is what the counter answers. R-Values are the ASCII values of the bytes in the answer.

You see that the GETVER answer is 14 bytes long, the GETVOLT answer only 1 byte (and the voltage is 4.2Volt), and GETCPS is answered with 2 bytes binary value (in newer counters it would be a 4 bytes binary value).

# Helpful External Software Tools

None is required to run GeigerLog, but sometimes certain tools can be really helpful.

## Editor Geany

Even if you don't want to edit the program code, you need a proper editor to adapt e.g. the Geiger-Log 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> .

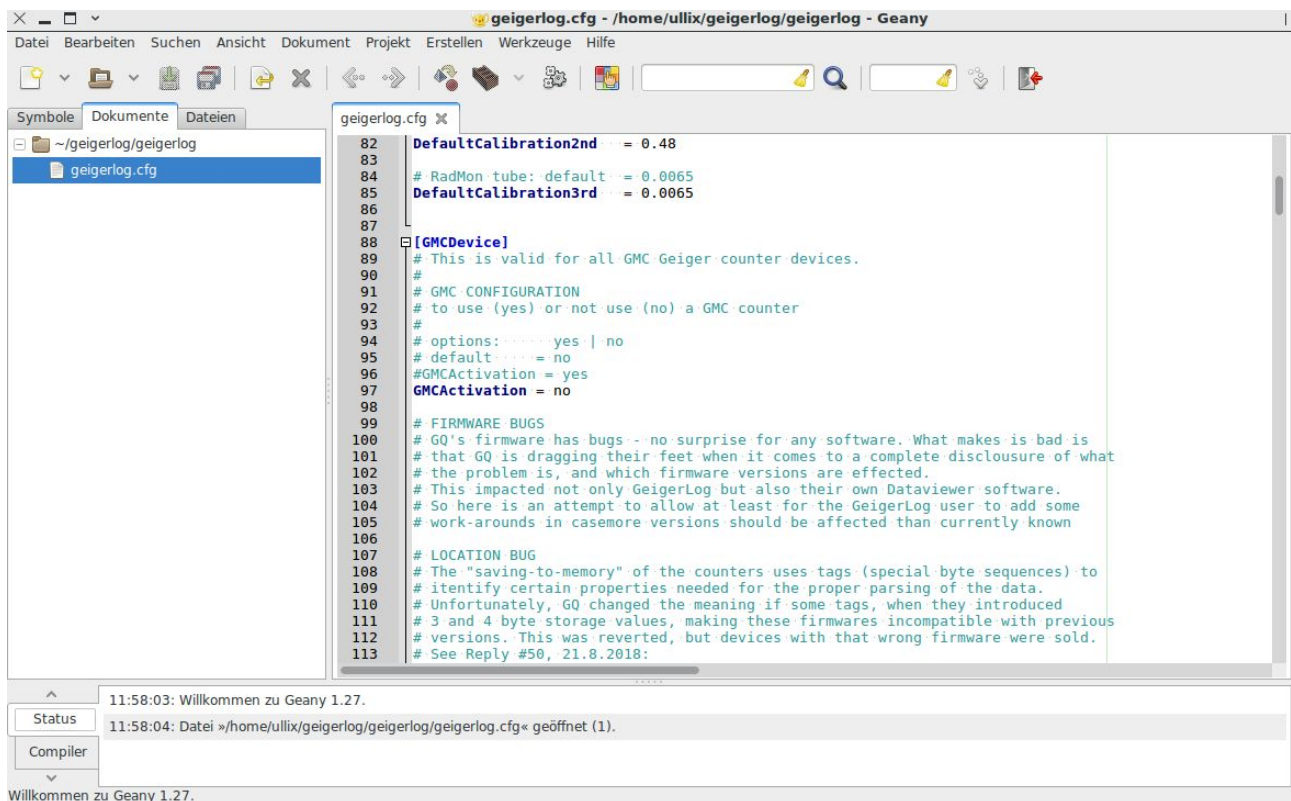


Figure 33: Editor Geany with file geigerlog.cfg opened

## DB Browser for SQLite

As the Log and History files are now saved as database files, you need a tool for inspecting and perhaps editing those files. The software **DB Browser for SQLite** is Open source and available for **Linux, Windows, and Mac**, and is an excellent tool for SQLite3 databases.

Download from <https://sqlitebrowser.org/> .

Viewing a Log database file in this browser may look like this:

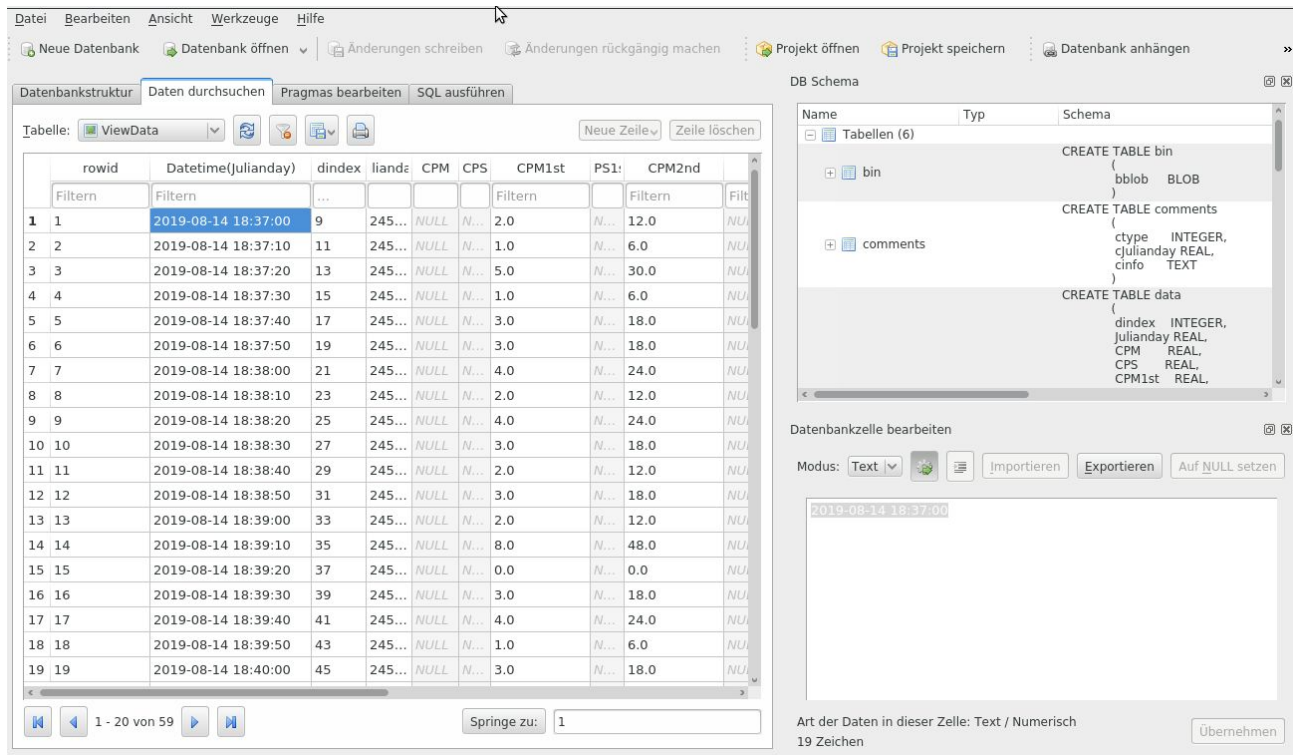


Figure 34: DB Browser for SQLite as tool to inspect Log files

# 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:

- [gmcmmap.com](http://gmcmmap.com) - This is the one supported by GQ Electronics
- [radmon.org](http://radmon.org) - Presently down after being hacked
- [safecast.org](http://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 contribute to [gmcmmap.com](http://gmcmmap.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  $\mu\text{Sv/h}$ . 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 ( $\mu\text{Sv/h}$ , or  $\text{nSv/h}$ ).

**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  $\text{CPM}=9999$ . 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,  $\mu\text{Sv/h}$  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 <sup>30)</sup>, 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 by 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  $\mu\text{Sv/h}$  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](#) <sup>31)</sup> provide only DAILY averages of quality controlled data!

## Occupational Radiation Limits

Available in menu **Help** → **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	<a href="#">BfS Grenzwerte</a>	<a href="#">OSHA</a>

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

---

30 it decreases with  $1/\sqrt{\text{count rate}}$ , see Poisson Distribution in [GeigerLog - Potty Training for Your Geiger Counter](#)

31 <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 100 for more guidance.

If you do encounter any bugs or problems please report via the project GeigerLog site at SourceForge: <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 Starting GeigerLog with Options on page 65.

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](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](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](http://nvlpubs.nist.gov/nistpubs/jres/42/jresv42n3p241_A1b.pdf)

**GQGMC**, Documentation by Phil Gillaspay

<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 even on the same version of the operating system!

## Style

Generally the default style will be ok (and it is markedly improved in the current PyQt5 toolkit over the previous PyQt4 toolkit). 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 <sup>32</sup>):

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

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

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

Exact spelling & capitalization is required!

## Fonts

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

---

<sup>32</sup> 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 Device and Computer using a Serial Connection

GeigerLog currently supports 3 devices, to which this chapter applies:

- GMC devices (Geiger counters)
- I2C devices (environmental sensors)
- Gamma-Scout devices (Geiger counters)

## BACKGROUND:

While device and computer are connected with a USB cable, their connection is actually only a classic serial connection, same as in the good old days of the teletype. The translation between USB and serial is done by an USB-to-Serial chip in the electronics of the device.

And while serial connections today are faster than decades ago, they are slow by today's standards. In the **GMC** and **I2C** devices the serial speed is in the order of 0.1 MBit/s <sup>33</sup>), while USB2 is nominal 480 Mbit/s and USB3.2 reaches 20 000 Mbit/s <sup>34</sup>)!

It is even worse with the **Gamma-Scout** devices, which support only 0.01 MBit/s or less <sup>35</sup>). But because they use a transfer algorithm of 7 bit with parity, the 8 bit bytes are split over a 2 bytes transmission, thereby reducing the effective transmission speed two-fold! Their latest model supports 460 800 baud, which, however, is also still reduced 2 fold, so no better than 0.2 MBit/s, i.e. only on par with the other devices.

So, none of these devices is anywhere close to a speed champion! With respect to logging or an occasional e.g. temperature measurement, the speed is sufficient, but for other actions, in particular downloading the history, a faster speed would be very welcome.

## The serial port, the baud rate, the driver, and permissions

For a successful connection you need to know the serial port's name given by the computer, the baud rate of the device, have a driver installed, and perhaps need the computer's permission to read and write to the port.

### Port names:

The port names on Linux are like '/dev/ttyUSB0', '/dev/ttyUSB1', ..., on Windows like 'COM3', 'COM17', ..., on Mac are less predictable, can be like '/dev/tty.USBSERIAL/', '/dev/tty.PL2303-xxx'. And worse: on unplugging and re-plugging the cable, the names may change!

### Baud rate:

For a **GMC** device one has to look up the baud rate directly at the counter by going 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

<sup>33</sup> 115 200 and 57 600 baud for GMC, up to 230 400 baud for the ELV dongle used for the I2c devices

<sup>34</sup> <https://www.elektronik-kompodium.de/sites/com/1310061.htm>

<sup>35</sup> 9 600 and 2 400 (!) baud

device, which seemed to have to do with the baud rate; and sometimes the counter chokes when things go too fast).

For an **I2C** device, the default baud rate is 115 200 baud, and can be doubled by software (not supported by GeigerLog; not worth for the very low single-value data rate).

For the **Gamma-Scout** devices the baud rate is fixed for a given device, and cannot be changed! It may be 2400, 9600, 460 800 baud. See the specs of the device <sup>36</sup>).

### **Driver:**

On Linux the driver is part of the system. On Windows and Mac a driver may have to be installed. Drivers may be available for download on the GQ website <sup>37</sup>); often a generic driver suffices.

### **Permissions:**

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.

On Windows 10 I have never had problems connecting to any of GeigerLog's device. I cannot give advice for a 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 device(s) with the computer and start GeigerLog. GeigerLog has two ways to help you finding the settings needed by your device(s):

- 1) showing you a list of ports detected at your computer, and allowing you to select ports and baudrate for your device(s)
- 2) doing an Auto-Discovery of the correct port and baudrate – Caution: this may have side effects, see below

### **1) Show & Select**

In the Help menu of GeigerLog choose **Show & Select USB Port and Baudrate**. A dialog pops up as shown in Figure 35. When you know which port belongs to which device and with which baud rate, you can make your choices, and press ok. GeigerLog will stop any logging, disconnect all of its devices, and reconnect them. If successful, everything is running.

These changes are only for the present run. If you restart, you have to make the choices again. Therefore, any settings that you know will remain permanent, you might want to enter into the configuration file `geigerlog.cfg`.

---

36 [https://www.gamma-scout.com/wp-content/uploads/Gamma-Scout\\_Communication\\_Interface\\_V1.7.txt](https://www.gamma-scout.com/wp-content/uploads/Gamma-Scout_Communication_Interface_V1.7.txt)

37 <http://www.gqelectronicsllc.com/comersus/store/download.asp>

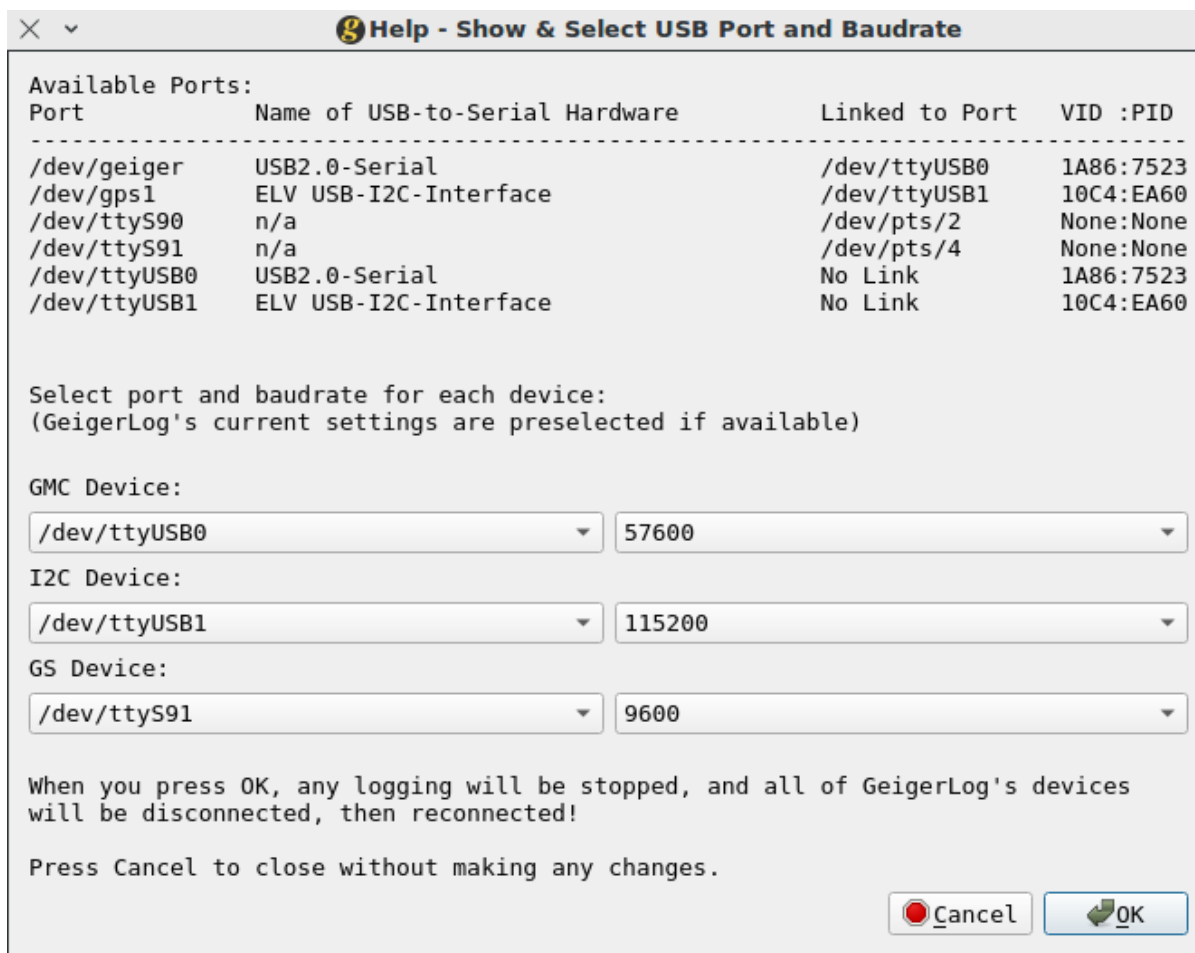


Figure 35: Show & Select Dialog for USB Port and Baudrate (result for a Linux system)

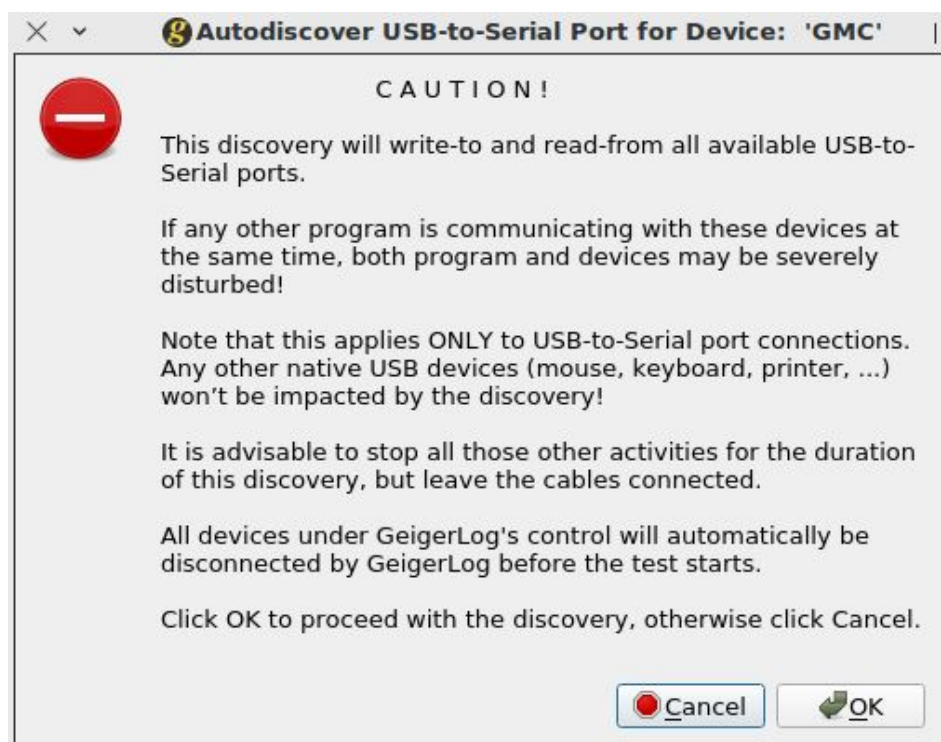


Figure 36: Autodiscovery USB-to-Serial Port Connection

## 2) Auto-Discovery

A significant disadvantage of serial connections is that it does neither tell you what device is connected, nor what its desired baudrate is. (It only tells you about the USB-to-Serial converter chip, not about the device connected to it!). Unless you know already, you have no choice but to write to each port with each possible baudrate with a message that would trigger a certain known response of only one of the devices if all parameters were correct, and you check if that response is given.

If only one device is connected this approach is straightforward and successful. However, if multiple devices are connected, a benign command for one device could run havoc with another device! At least it might get disturbed enough that you have to restart the device, or need a factory reset.

Removing all cables and leaving only one device connected for the test is also no option, because the computer will likely re-associate the ports after re-connection, and the names are bound to be different.

When you start the auto-discovery from menu **Help** → **Autodiscover USB Port for Device:** → **<your device>** the warning shown in Figure 36 pops up. Make very sure you follow its advice!

It is not easy using a USB-to-Serial Port with multiple devices!

After the auto-discovery has run – may take several 10 sec – the result may look like in Figure 37 shown for a Linux system.



Figure 37: USB Autodiscovery result

This tells you that device 'GS' (a Gamma-Scout counter) was found at Port: /dev/ttyttyS91 with Baudrate: 460 800, while the current setting of GeigerLog (bottom) is: Port: /dev/ttyS91 and Baudrate: 9 600.

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

To permanently use the just found USB port parameters, edit the section **[XYZSerialPort]**, where XYZ stands for the respective device (GMC, I2C, or GammaScout) of the configuration file 'geigerlog.cfg' and save, and then restart GeigerLog (modified lines in red) :

```
[GammaScoutSerialPort]
# default = /dev/ttyUSB2
port      = /dev/ttyS91

# the device must have been set to the selected baudrate!
# default = 9600
baudrate  = 460800
```

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, ... and the configuration file modified to:

```
port      = COM3
baudrate  = 460800
```

When running GeigerLog you can get a brief info on setting and using 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 a device to the USB port on a Linux system, 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 a device 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. Entering:

```
groups myname
```

results in:

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

This change is permanent; it 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. I have marked the two USB-to-Serial adapter 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.ggelectronicsllc.com/forum/topic.asp?TOPIC\\_ID=4447](http://www.ggelectronicsllc.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/gggmc/files/gggmc/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.ggelectronicsllc.com/forum/topic.asp?TOPIC\\_ID=4948](http://www.ggelectronicsllc.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 73).

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.ggelectronicsllc.com/forum/topic.asp?TOPIC\\_ID=4948](http://www.ggelectronicsllc.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 Gillaspay in this document <https://SourceForge.net/projects/gqgmc/files/gqgmc/GQ-GMC-ICD.odt/download>. Other info comes from a reverse engineering 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 2<sup>nd</sup> 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) <sup>38)</sup>.

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

<sup>38)</sup> 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. When 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 lose 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.

## All Parsing Problems are Encountered by All Software!

Just to be sure: these are problems created by poor firmware! All software, including GQ’s own software, is impacted by these firmware issues!

# 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 claims 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.ggelectronicssl.com/forum/topic.asp?TOPIC\\_ID=4948](http://www.ggelectronicssl.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  $(\text{hex}1000 \text{ AND } 0\text{FFF} + 1) = 1$  data point instead of 4096. The workaround is to request  $4096 - 1 = 4095$  data points, which results in  $(\text{hex}0\text{FFF AND hex}0\text{FFF}) = \text{hex}0\text{FFF}$ , 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 82), 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.ggelectronicsllc.com/forum/topic.asp?TOPIC\\_ID=4948](http://www.ggelectronicsllc.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 by GeigerLog 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

“Calibration” seems to be an easy concept, but there are some problems.

What we measure with a Geiger counter is a ‘count’. And we typically determine the count **rate** like CPM, or CPS, or counts per any other time duration, like hours (CPH), days (CPD), etc..

But what we are really interested in is not the technical number count rate, but the health effect to be expected from that count rate.

Now imagine you take a bunch of Geiger counters, using same or different Geiger tubes, with the same or different operating conditions (in particular anode voltage) and measure one unknown radioactive source. You will get different to very different CPM values, like – see below – CPM=2 to CPM=154! On what grounds do you decide that the source is harmless or a severe threat?

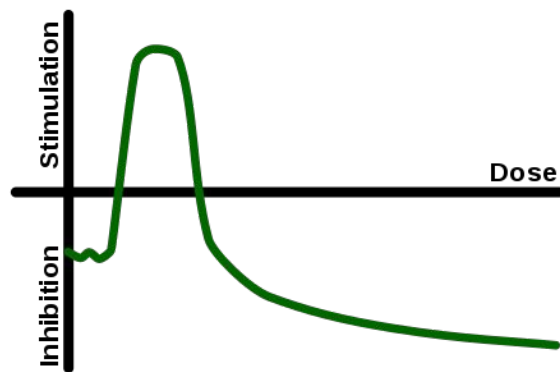
You probably would want first to equalize the readings from the counters. That is rather easy: you take a standardized radioactivity source, call one of the counters your reference, and determine correction factors for all others. So, all counters now give the same answer, but you still don’t know what the health effect is.

Fortunately there is a lot of research on the health topic, and there are standards, which are characterized also with respect to health issues. Now you can not only equalize the readings of all counters, but convert the count rates to an absolute health-relevant factor, and this is our desired calibration factor.

The health effect of **a count** is quantified as a **dose** in units of Sv, which stands for ‘**Sievert**’<sup>39</sup>), named after the Swedish physicist Maximilian Sievert. The health effect of a count **rate** in CPM is most commonly converted to a dose **rate** in  $\mu\text{Sv/h}$  (Micro-Sievert per hour).

## Notes:

- A count rate like CPM is obviously also a dose rate, just reported in a different unit.
- In converting the units, a **linear** relationship between dose rate in CPM and Dose rate in  $\mu\text{Sv/h}$  is implied! This assumption is anything but trivial, because we know e.g. from chemicals the effect called **hormesis**<sup>40</sup>), which means that a substance can not only have a strongly non-linear dose effect, but also that – simply said – too little may be as bad as too much. (Graph from Wikipedia)



Think e.g. of table salt: too much kills you, too little also!

For radioactivity the assumption of linearity is based on the LNT (Linear No-Threshold)<sup>41</sup>)

39 <https://en.wikipedia.org/wiki/Sievert>

40 <https://en.wikipedia.org/wiki/Hormesis>

41 [https://en.wikipedia.org/wiki/Linear\\_no-threshold\\_model](https://en.wikipedia.org/wiki/Linear_no-threshold_model)

theory, which became the basis for all radioactivity protection regulations worldwide!  
Hormesis in radioactivity is explicitly excluded in this model! This is hotly debated, but it became the norm. And it is implied when using a **single, constant** Calibration Factor independent of the rate.

The Calibration Factor as now defined in GeigerLog is:

**This Calibration Factor is the inverse of previous definitions! See next paragraph!**

**Dose Rate [ $\mu\text{Sv/h}$ ] =  $1 / \text{Calibration Factor [CPM / ( $\mu\text{Sv/h}$ )]} * \text{Dose Rate [CPM]}$**

or

**Dose Rate [CPM] =  $\text{Calibration Factor [CPM / ( $\mu\text{Sv/h}$ )]} * \text{Dose Rate [ $\mu\text{Sv/h}$ ]$**

## GeigerLog 1.0 uses a new definition for the calibration factor

Before Version 1.0 GeigerLog followed the definitions used by GQ for their tubes. This changed with this release of GeigerLog. The reason is that the new definition allows much easier comparisons of tube sensitivities: higher numbers mean higher sensitivity.

Let's do a quick test:

you are given these calibration factors old-style of 0.0065, 0.48, 0.002637, 0.00926, 0.09 and 0.42. Quick, sort them by sensitivity from high to low, and tell by how much the first tube is more sensitive than the last!

Isn't that a lot easier when you get calibration factors new-style of 154, 2.08, 379, 108, 11.1, and 2.38?

To give numeric examples using various tubes and sensors:

	Old	New	
Tube/Sensor	[ $\mu\text{Sv} / \text{h} / \text{CPM}$ ]	[ $\text{CPM} / ( \mu\text{Sv} / \text{h} )$ ]	
M4011:	0.0065	<b>154</b>	$1 / 0.0065 = 153.85$
SI3BG:	0.48	<b>2,08</b>	$1 / 0.48 = 2.083$
LND 7317	0.002637	<b>379</b>	$1 / 0.002637 = 379.22$
LND 712:	0.00926	<b>108</b>	$1 / 0.00926 = 107.99$
SGP-001:	0.09	<b>11,1</b>	$1 / 0.09 = 11.111$
SI3BG (Synch)	0.42	<b>2,38</b>	$1 / 0.42 = 2.381$

For use in GeigerLog the new numbers are rounded to **3 significant** decimals.

## The Dilemma

A calibration factor will be determined with a specific setup, and is valid **ONLY** for the conditions used in the setup.

While this seems trivial, it is largely ignored when it comes to Geiger counters.

What setup is being used for determining the calibration factor for the M4011 tube in a GMC Geiger counter? Well, we don't know!

To my knowledge, GQ has never reported how the calibration factor had been determined. Nor has the manufacturer – it isn't even known who the manufacturer is – of this tube ever released a datasheet. If it exists at all, it has not been made public. So, which calibration factor can be used?

## Looking Elsewhere for Specs

There is a SBM20 tube, an old Russian Geiger tube, similar in shape to the M4011, albeit it is made from steel, not from glass. And for the SBM20 one does find specifications, like here: <http://www.gstube.com/data/2398/>

Gamma Sensitivity Ra <sup>226</sup> (cps/mR/hr)	29
Gamma Sensitivity Co <sup>60</sup> (cps/mR/hr)	22

The units are different from what we use, but we can make these arguments: Co60 is a beta and gamma emitter; Ra226 is an alpha, beta and gamma emitter. However, for calibration standard purposes both are typically packaged such that only gamma can escape the package, and if so we can assume pure gamma emission. With that we can equate mR with mRem, and with 1 mRem = 10μSv, we get:

Calibration Factor	new-style	old-style						
Ra226: 29 * 60 / 10 =	174 CPM / ( μSv / h )	0.0058	μSv	/	h	/	CPM	
Co60: 22 * 60 / 10 =	132 CPM / ( μSv / h )	0.0076	μSv	/	h	/	CPM	
Average of the two:	153 CPM / ( μSv / h )	0.0067	μSv	/	h	/	CPM	
GQ's M4011 calibration:	154 CPM / ( μSv / h )	0.0065	μSv / h	/	CPM			

GQ's calibration is strikingly close to the average of the two, and with nothing better at hand we'd say that this is the base for GQ's calibration factor. And perhaps it is the sole base, we don't know.

Thus, when the two tubes are directly compared, they should give the same results. Both tubes can be run with the same voltage, and the SBM20 can even be used instead of the M4011 in the GMC counters. I used the M4011 and the SBM20 in an GMC-300E+ counter and published the results <sup>42</sup>).

As shown in Figure 38 the SBM20 background is 3% higher than the M4011 background. However, this seems to be well within statistical uncertainty, and the conclusion is that the background is not different.

When measured with the radioactive sources Potassium and Thorium, the SBM20 shows elevated counts of 41% ... 45%, as shown in Figure 39.

However, as we started with the assumption that the SBM20 calibration has been made with pure gamma emitters, any comparison of the tubes can only be made with pure gamma emitters. But both K and Th are strong beta emitters.

Therefore from the experiment shown in Figure 39 we can **NOT** draw the conclusion that the SBM20 is more sensitive **to gamma** than the M4011; the question remains open.

42 [http://www.ggelectronicssl.com/forum/topic.asp?TOPIC\\_ID=4571](http://www.ggelectronicssl.com/forum/topic.asp?TOPIC_ID=4571)



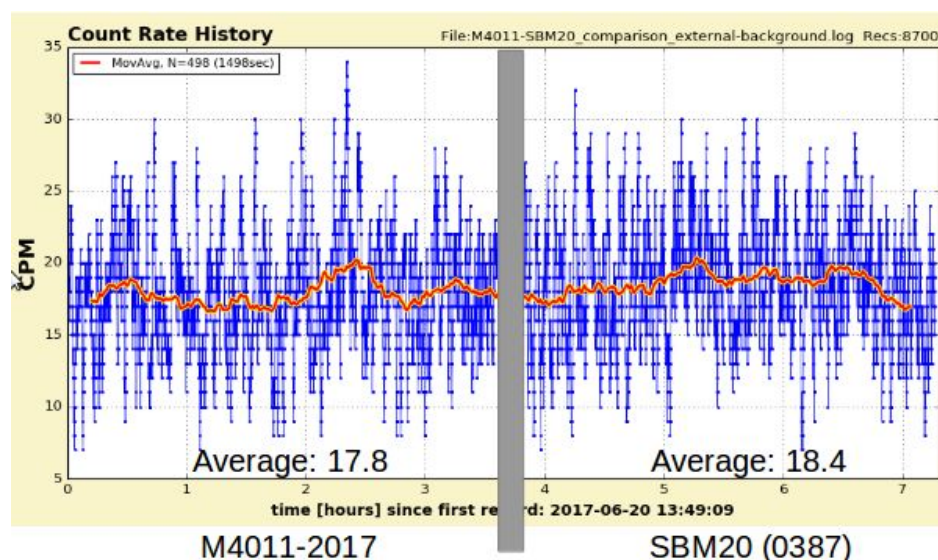


Figure 38: Comparison of a M4011 and SBM20 tube inserted in a GMC-300E+ counter measuring Background

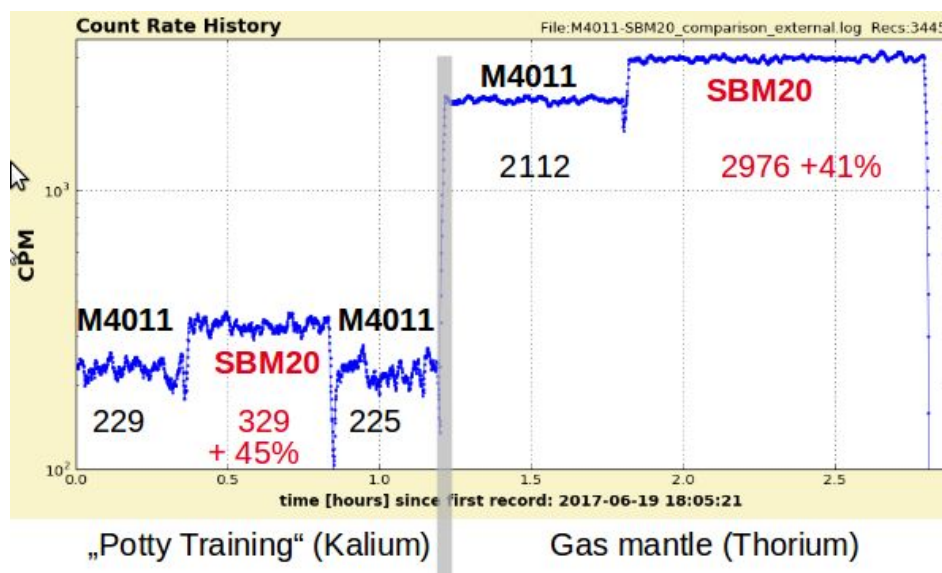


Figure 39: Comparison of a M4011 and SBM20 tube inserted in a GMC-300E+ counter measuring a Potassium and Thorium source, resp.

In extension, we also have to assume that the calibration for a M4011 is only for pure gamma emitters, and NOT for beta emitters. We simply do not know what the calibration factor is for beta!

And likewise, using a calibration factor with unknown specification, and likely relevant for gamma only, on tubes with sensitivity for alpha when measuring alpha radiation, is simply additional non-sense! When you measure any alpha radiation, report the CPM and describe tube used and setup. But do not report dose rate in  $\mu\text{Sv/h}$ , because that tells the reader that you did not understand what you were doing!

## The Energy Dependence of the Calibration Factor

Looking at the gamma spectra in fig. G1 we see that Co60 is strong above 1 MeV, while most Ra226 is below 0.5 MeV. The SBM20 tube, according to its specs, is 32% more sensitive to the lower 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.

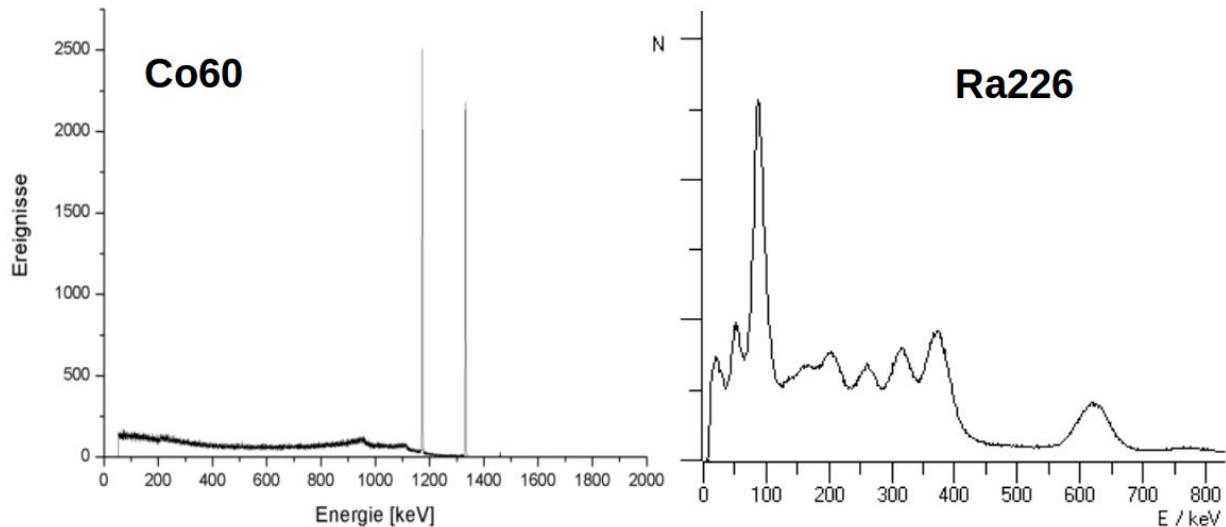


Fig. G1 Gamma Spectra of Ra226 and Co60

As there is no info at all available for the M4011 tube, there are also no energy dependence data available. But also for other tubes and gamma energies, the data are scarce. On Wikipedia <sup>43)</sup> one finds this “generic” graph for an energy dependence of a Geiger-Müller tube:

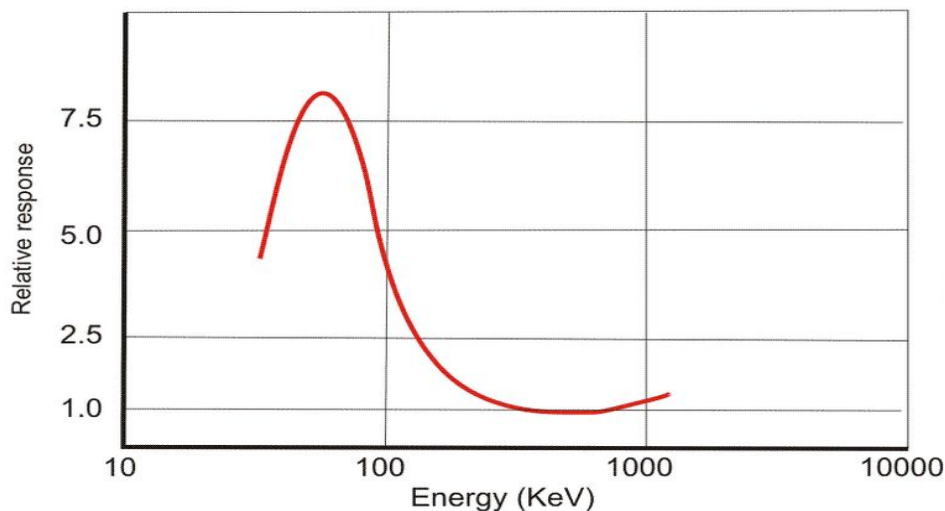


Figure 40: Relative Response of a Geiger Tube with Respect to Gamma Energy

Therefore the calibration factor is not valid for all gamma rays, but **ONLY** for that mix of gamma rays the calibration factor was initially determined with, which, however, is unknown!

43 [https://en.wikipedia.org/wiki/Geiger%E2%80%93M%C3%BCller\\_tube](https://en.wikipedia.org/wiki/Geiger%E2%80%93M%C3%BCller_tube)



## Summary

1. The Calibration Factor is valid **ONLY** for the type of radioactivity source, for which it had initially been calibrated.
2. It is **unknown** which calibration setup – if any – had ever been used for the M4011 tube
3. The Calibration Factor is **NOT** valid for a beta emitter
4. The Calibration Factor is **NOT** valid for a alpha emitter

This does not imply that you can't measure anything else but the original calibration source. But you have to specify, which calibration factor you used, and how you determined it.

In premium Geiger counters you have the option of applying a calibration factor based on C060, Ra226, or Cs137.

## Tuning the counter

Since the case of the counter is basically transparent to at least higher energy 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!

## The GMC counter calibrations

The GQ GMC counters have 3 calibration points, which would allow to accommodate some non-linearity. However, since the LNT theory has to be applied, there is no non-linearity to be considered! In theory the three points could also be used to correct some dead-time effects at very high count rates. However, that would involve some complicated calculations, as you would not only have to take the dead-time correction into account, but also the shift in the calibration due to that correction.

The proper way of doing that is already implemented in GeigerLog, see the **Formula Interpreter** in chapter Scaling on page 27. A broader discussion can also be found in this GQ forum topic: [https://www.ggelectronicsllc.com/forum/topic.asp?TOPIC\\_ID=5357](https://www.ggelectronicsllc.com/forum/topic.asp?TOPIC_ID=5357); in particular look into Reply #9 and Reply #33.

But, this discusses a moot point anyway, because all 3 points in all GMC counters establish the same slope, hence effectively only a single calibration point is used. This is what GeigerLog reads out from the GMC counters:

```
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)
```

But be aware of counters where this wrong calibration is read out: <sup>44</sup>).

Device Calibration:

Calibration Point 1:	60 CPM =	0.39 $\mu\text{Sv/h}$	(0.0065 $\mu\text{Sv/h}$ / CPM)
Calibration Point 2:	10000 CPM =	65.00 $\mu\text{Sv/h}$	(0.0065 $\mu\text{Sv/h}$ / CPM)
Calibration Point 3:	25 CPM =	9.75 $\mu\text{Sv/h}$	(0.3900 $\mu\text{Sv/h}$ / CPM)

GeigerLog now uses a default calibration of 154 CPM / ( $\mu\text{Sv/h}$ ) (old value: 0.0065  $\mu\text{Sv/h}$  / CPM) for the M4011 tube, and 379 CPM / ( $\mu\text{Sv/h}$ ) (old value: 0.002637  $\mu\text{Sv/h}$  / CPM) for the LND 7317 tube in the GMC-600+.

But all calibration factors can be changed, temporarily in GeigerLog during a run, and more permanently in GeigerLog's configuration file.

## Calibration Factor for the Low-Sensitivity Tube SI3BG

Forum user Ikerrg has thankfully contributed these data. They were generated with the double-tube counter GMC-500+ exposed to data from a Synchrotron <sup>45</sup>). This is a device which can also generate strong gamma radiation.

The counter was held into the radiation field, and the counts recorded into its history buffer. The history was read out with GeigerLog and analyzed, as shown in Figure 41.

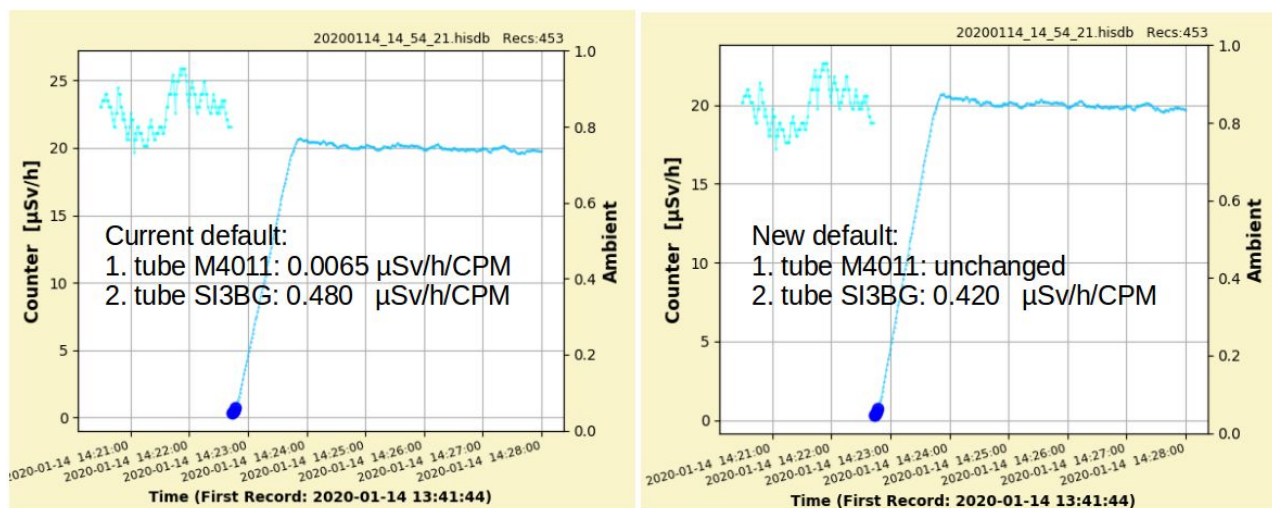


Figure 41: Synchrotron radiation captured with a GMC-500+ counter, and analyzed with GeigerLog

Note the counter axis is not in CPM but in  $\mu\text{Sv/h}$ ! Data colored in Cyan are recorded with the 1<sup>st</sup> tube M4011, using its calibration factor 154 CPM / ( $\mu\text{Sv/h}$ ) (old style: 0.0065  $\mu\text{Sv/h}$ /CPM). Then the tubes were switched and recording continued with the 2<sup>nd</sup> tube colored sky-blue. Relative to the 1<sup>st</sup> tube the dose rate of the 2<sup>nd</sup> comes out too low (left picture). This is corrected when the Calibration Factor of the 2<sup>nd</sup> is changed from (old-style) 0.48 to 0.42  $\mu\text{Sv/h}$ /CPM. In new-style this is a change from 2.08 to 2.38 CPM / ( $\mu\text{Sv/h}$ ) (right picture).

<sup>44</sup> GMC-500 counters were delivered with this calibration setting

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 found in a GMC-500, which has **no** second tube.

<sup>45</sup> <https://en.wikipedia.org/wiki/Synchrotron>

But, as we don't know the energy profile of the Synchrotron radiation, nor do we know for what the original calibration was for – nor if there ever was one – we don't know if that new value is any better or not than the old one.

And anyway, it is only relative to the M4011 tube.

But at least it gives us a ballpark feeling that the 2<sup>nd</sup> tube is  $154 / 2.08 \dots 154 / 2.38$ , or 74 ... 65 times less sensitive.

# 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 – the Proper Version is Python 3.8

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, 3.7, and 3.8, both on Linux and on Windows 10.

Python version 3.5 and earlier are now past their end-of-life status and should no longer be used for new projects. The latest Python is currently (Oct 2020) version 3.9.0, which however is not yet usable for GeigerLog, because the very essential module matplotlib cannot be installed!

**Python 3.8** is therefore now the preferred version.

If you have an older version already installed, just try it – it may work. But errors and bugs will only be investigated when 3.8 or later is used.

## 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** <sup>46</sup>) 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 not needed

The default installation of Python will typically for all users, and has to be installed with administrative rights. However, the Pip installed packages will be installer per user; so administrative rights will NOT be needed!

## 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 ‘python3’.

---

<sup>46</sup> Pip is a recursive acronym that can stand for either "Pip Installs Packages" or "Pip Installs Python". Homepage: <https://pypi.org/project/pip/>

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

```
python -V
python2 -V
python3 -V
```

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

We will now assume that both Python is installed as version 3, and you have to use the command **'python3'** to start your Python 3. 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 Python3 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.

Required by GeigerLog, and latest versions as of October 2020, verified to work with GeigerLog Version 1.0, are:

- |               |                              |
|---------------|------------------------------|
| • pip         | latest version: 20.2.3       |
| • setuptools  | latest version: 50.3.0       |
| • PyQt5       | latest version: 5.15.1       |
| • PyQt5-sip   | latest version: 12.8.1       |
| • matplotlib  | latest version: 3.3.2        |
| • numpy       | latest version: 1.19.2       |
| • scipy       | latest version: 1.5.2        |
| • matplotlib  | latest version: 3.3.2        |
| • pyserial    | latest version: 3.4          |
| • paho-mqtt   | latest version: 1.4.0        |
| • sounddevice | latest version: 0.4.1        |
| • SoundFile   | latest version: 0.10.3.post1 |

Not required by GeigerLog, but recommended for installation:

- |             |                     |
|-------------|---------------------|
| • pip-check | latest version: 2.6 |
|-------------|---------------------|

Older and newer versions will work most of the time, but when you experience problems, make sure to have the listed versions installed! Guidance for installing a specific version is found in next paragraph.

## 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! Some advanced use of Pip is explained in chap-

ter Appendix I – Advanced Use of Pip on page 110, here is some brief basic usage using the installation of package numpy as example:

To do a fresh install of package numpy, simply enter in a command window:

```
python3 -m pip install numpy
```

If numpy is already installed, check the version:

```
python3 -m pip show numpy
```

If an upgrade is needed, do it with option -U:

```
python3 -m pip install -U numpy
```

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

```
python3 -m pip 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’, verbose ‘-v’, 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/>.

## Inspecting your files

As the Log and History files are now saved as database files, you need a tool for inspecting those files. See my recommendation in chapter Helpful External Software Tools on page 69.

# Linux – Installation

This was tested with **Ubuntu Mate 16.04.6 LTS, 18.04 LTS, 19.04**, and also shown to run on **Mint**.

## Installation of Python and Pip

Install Python Version 3.8. If the distribution does not offer this by default, you may have to activate a suitable repository. On Ubuntu-like systems this is the **deadsnakes** repository (<https://github.com/deadsnakes>), which provides many new and old versions of Python:

```
sudo add-apt-repository ppa:deadsnakes/ppa)
```

If Python 3.8 is not available, try the next lower version.

Install :

```
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.

The `'-U'` at each command ensures that the most recent version of each module will be installed even if an older version is already installed:

```
python3 -m pip install -U pip
```

Now install all other packages:

```
python3 -m pip install -U setuptools
python3 -m pip install -U PyQt5
python3 -m pip install -U PyQt5-sip
python3 -m pip install -U matplotlib
python3 -m pip install -U numpy
python3 -m pip install -U scipy
python3 -m pip install -U pyserial
python3 -m pip install -U paho-mqtt
python3 -m pip install -U sounddevice
python3 -m pip install -U soundfile
python3 -m pip install -U pip-check
```

Installing `pip-check` is optional.

List all installed packages and verify their versions (for references see Other Packages on page 101)

```
python3 -m pip 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 into the `geigerlog` folder and start GeigerLog from the terminal with:

```
./geigerlog
```

Or start with:

```
python3 /path/to/geigerlog
```

## Installation Problems

Note: Special thanks to user theMike!

Some Linux distributions may require to explicitly install additional programs from their repository. Candidates are:

```
sudo apt-get install python3-setuptools
sudo apt-get install python3-dev
```

Sometimes a module installed by the distribution 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 Mint.)

An obscure error message resulted from the use of PyQt5 version 5.15.0 in Python 3.6:

```
qt.qpa.plugin: Could not load the Qt platform plugin "xcb"
in "" even though it was found.
This application failed to start because no Qt platform plugin
could be initialized. Reinstalling the application may fix this
problem.
```

This is very misleading. The solution is to install **libxcb-xinerama0** from the repository:

```
sudo apt-get install libxcb-xinerama0
```

## Installing Multiple Versions of Python

Installing multiple Python versions from the repository is easily possible, but once you start adding packages using Pip, you are almost guaranteed to run into severe problems.

The solution is to use a virtual environment. This is for Python-only, it is NOT a virtualization of your whole system, and it is quite simple!

Follow the guidance in this document: “**GeigerLog-HOWTO use Python in a virtual Environment on Linux**” found among the GeigerLog articles on the SourceForge site <sup>47</sup>).

This also allows to configure variants of a specific Python version, e.g. using different Pip installed modules.

---

<sup>47</sup> <https://sourceforge.net/projects/geigerlog/files/Articles/>



# Windows - Installation

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

## Installation of Python and Pip

Preferably install Python Version 3.8. Download from: <https://www.python.org/downloads/windows/>.

You want the “**Windows x86-64 executable installer**” :

<https://www.python.org/ftp/python/3.8.6/python-3.8.6-amd64.exe>

Open it **with administrative rights!**

### In the installer:

- check: add Python 3.8 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

Remember to verify (see chapter Verify the current Python and Pip installation status on your machine on page 100) what your command is to call Python in version 3, I continue to assume it is python3; your command may be a simple ‘python’.

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

The ‘-U’ at each command ensures that the most recent version of each module will be installed even if an older version is already installed. The - -user option will install the package only for the current user:

```
python3 -m pip install --user -U pip
```

Now install all other packages:

```
python3 -m pip install --user -U setuptools
python3 -m pip install --user -U PyQt5
python3 -m pip install --user -U PyQt5-sip
python3 -m pip install --user -U matplotlib
python3 -m pip install --user -U numpy
python3 -m pip install --user -U scipy
python3 -m pip install --user -U pyserial
```

```
python3 -m pip install --user -U paho-mqtt
python3 -m pip install --user -U sounddevice
python3 -m pip install --user -U soundfile
python3 -m pip install --user -U pip-check
```

Installing pip-check is optional.

List all installed packages and verify their versions (for references see Other Packages on page 101)

```
python3 -m pip 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

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 Microsoft website:

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

## 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/> Download the latest version of 3.8. Preferably use 64 bit.

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 100.

### Using pip

Once pip for Py3 is installed, upgrade it first:

```
python3 -m pip install --user -U pip
```

Now install all other packages:

```
python3 -m pip install --user -U setuptools
python3 -m pip install --user -U pyqt5
python3 -m pip install --user -U pyqt5-sip
python3 -m pip install --user -U matplotlib
python3 -m pip install --user -U numpy
python3 -m pip install --user -U scipy
python3 -m pip install --user -U pyserial
python3 -m pip install --user -U paho-mqtt
python3 -m pip install --user -U sounddevice
python3 -m pip install --user -U soundfile
python3 -m pip install --user -U pip-check
```

Installing pip-check is optional.

List all installed packages and verify their versions (for references see Other Packages on page 101)

```
python3 -m pip 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
```

# Raspberry Pi Installations

I tested this on a **Raspi4** using its default Raspian installation based on **Buster**, the latest version of Debian Linux. So this is a Linux installation, and everything said for Linux should work here.

However, there are a few catches, as I had found out.

Some of the software needed by GeigerLog is already installed, and, in contrast to what I emphasized for all other installation, you better do NOT change it by using Pip! And more, whenever anything can be installed by apt, it should NOT be installed or updated by Pip! And even more, other software is installed and usable, but cannot be detected by Pip. This applies to PyQt5 and PyQt5-sip!

Very strange.

Here the summary:

Python3	installed by default
Pip	installed by default
PyQt5	installed by default (not seen by Pip)
PyQt5-sip	installed by default (not seen by Pip)
numpy	installed by default
matplotlib	installed by default
scipy	install via Add/Remove software installer
paho-mqtt	install via Add/Remove software installer
soundfile	install via Add/Remove software installer
libffi-dev	install via Add/Remove software installer
geany	installed by default

The one exception to the warning to not upgrade anything with Pip is Pip itself. Do upgrade that with the standard

```
python3 -m pip install --user -U pip
```

And then Pip-install:

```
python3 -m pip install --user -U sounddevice
python3 -m pip install --user -U pip-check
```

And finally, when you use the Raspi with GeigerLog to read an AudioCounter – you need a USB-Soundcard for this as the Raspi has no audio input – make sure to select the USB-Soundcard as audio device, and **enable Capture** in the sound settings!

Raspi can perform using the AudioCounter driver via a USB-Soundcard, but it has its limits. That small computer is simply not powerful enough!

I bought this device <https://www.amazon.de/gp/product/B07RS11PDD> (8.18Euro) which has this chip: 'ID 1b3f:2008 Generalplus Technology Inc.' Works equally well on Raspi and on Desktop; just plug it in and select it as your audio device, no drivers needed.



Then I connected a GMC300E+ counter simultaneously both via its digital USB-to-Serial output and via its Audio output to the computer. The computer running GeigerLog was either the Raspi or my Desktop.

The summary of results is shown in this picture:

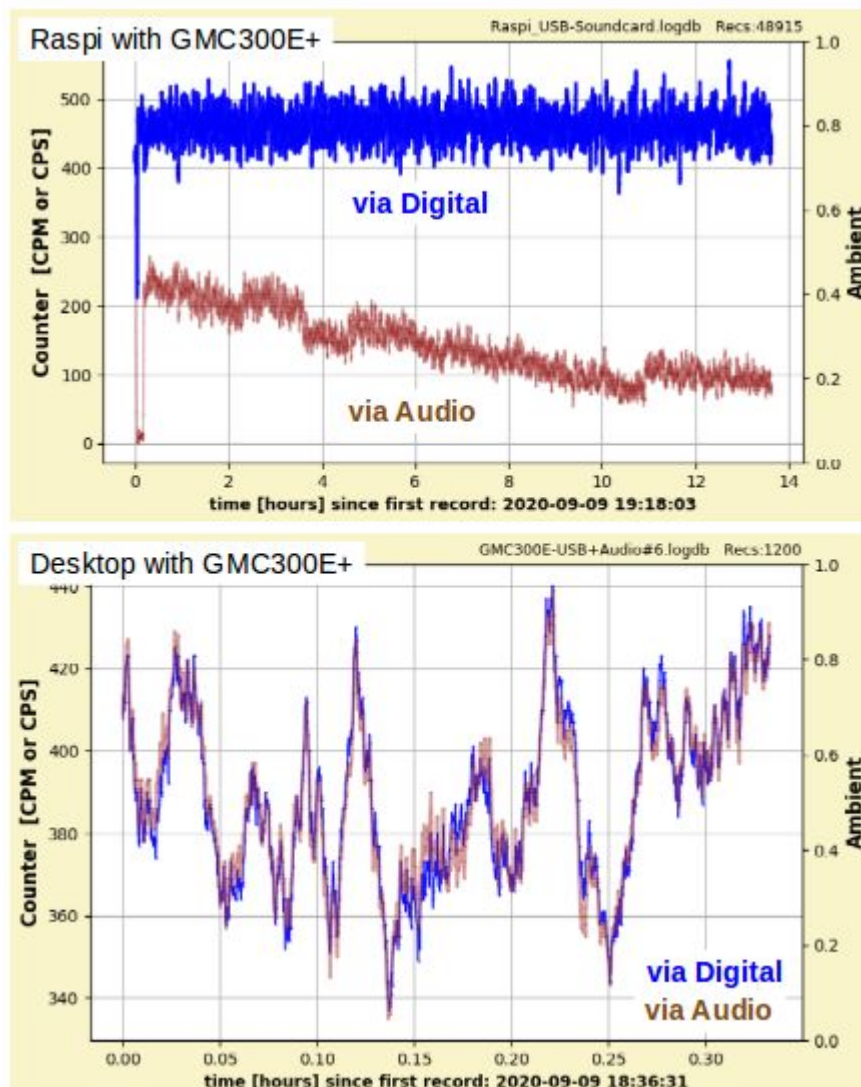


Figure 42: Using a Raspberry Pi with GeigerLog

The upper one shows Raspi results: while Raspi easily handles GeigerLog with a digital connection to a GMC counter, it is simply overwhelmed with the audio processing.

The lower pic shows the very same setup, but now connected to my Desktop. One can barely see a load on the CPU. The data are identical within statistical scatter. The main reason for differences is that the time base is not and cannot be synchronized. In this particular example the overall average is Digital-CPM=388.48, while the Audio-CPM=388.45 ;-)) Typically the differences are about 1%.

Conclusion: Raspi is perfectly fine for Digital, but not much useful as AudioCounter!

# 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:

```
python3 -m pip list --outdated
```

## Showing detailed version information of a specific package

To show details on e.g. matplotlib enter:

```
python3 -m pip show matplotlib
```

The output will be like:

```
Name: matplotlib
Version: 3.3.2
Summary: Python plotting package
Home-page: https://matplotlib.org
Author: John D. Hunter, Michael Droettboom
Author-email: matplotlib-users@python.org
License: PSF
Location: /home/ullix/geigerlog/vgl38/lib/python3.8/site-packages
Requires: certifi, kiwisolver, pillow, cyclor, numpy, python-dateutil, pyparsing
Required-by:
```

## 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):

```
python3 -m pip install --force-reinstall paho-mqtt==1.3.1
```

## Looking for a specific version

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

```
python3 -m pip 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: GLpipcheck

GLpipcheck comes as part of GeigerLog; you find it in the GeigerLog directory. It tells you which packages required by GeigerLog are installed or are missing, and which ones have upgrades available. Start with:

```
python3 /path/to/GLpipcheck
```

See an example output here: GLpipcheck on page 66.

## 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 2.6 (Oct 2020).

Install or update pip-check:

```
python3 -m pip install -U 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	<a href="https://pypi.python.org/pypi/Glances">https://pypi.python.org/pypi/Glances</a>
influxdb	2.12.0	5.2.1	<a href="https://pypi.python.org/pypi/influxdb">https://pypi.python.org/pypi/influxdb</a>
keyring	7.3	18.0.0	<a href="https://pypi.python.org/pypi/keyring">https://pypi.python.org/pypi/keyring</a>
Minor Release Update	Version	Latest	
bottle	0.12.7	0.12.16	<a href="https://pypi.python.org/pypi/bottle">https://pypi.python.org/pypi/bottle</a>
cffi	1.11.5	1.12.2	<a href="https://pypi.python.org/pypi/cffi">https://pypi.python.org/pypi/cffi</a>
cloudpickle	0.6.1	0.8.0	<a href="https://pypi.python.org/pypi/cloudpickle">https://pypi.python.org/pypi/cloudpickle</a>
cryptographv	2.3.1	2.6.1	<a href="https://pypi.python.org/pypi/cryptographv">https://pypi.python.org/pypi/cryptographv</a>
Unchanged Packages	Version	Latest	
LabJackPython	2.0.0	2.0.0	<a href="https://pypi.python.org/pypi/LabJackPython">https://pypi.python.org/pypi/LabJackPython</a>
matplotlib	3.0.3	3.0.3	<a href="https://pypi.python.org/pypi/matplotlib">https://pypi.python.org/pypi/matplotlib</a>
networkx	2.2	2.2	<a href="https://pypi.python.org/pypi/networkx">https://pypi.python.org/pypi/networkx</a>
numpy	1.16.2	1.16.2	<a href="https://pypi.python.org/pypi/numpy">https://pypi.python.org/pypi/numpy</a>
paho-mqtt	1.4.0	1.4.0	<a href="https://pypi.python.org/pypi/paho-mqtt">https://pypi.python.org/pypi/paho-mqtt</a>
photocollage	1.4.4	1.4.4	<a href="https://pypi.python.org/pypi/photocollage">https://pypi.python.org/pypi/photocollage</a>
pip	19.0.3	19.0.3	<a href="https://pypi.python.org/pypi/pip">https://pypi.python.org/pypi/pip</a>
pip-check	2.3.3	2.3.3	<a href="https://pypi.python.org/pypi/pip-check">https://pypi.python.org/pypi/pip-check</a>
pyalsaudio	0.8.4	0.8.4	<a href="https://pypi.python.org/pypi/pyalsaudio">https://pypi.python.org/pypi/pyalsaudio</a>
PyAudio	0.2.11	0.2.11	<a href="https://pypi.python.org/pypi/PyAudio">https://pypi.python.org/pypi/PyAudio</a>

Figure 43: 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/>. #
#####
```