GeigerLog Manual

by ullix

Version 1.3

January 2022

What’s New in GeigerLog 1.3 ?

- **WiFi enablement** of GeigerLog is the most significant change in this release!

  You can now **monitor GeigerLog from your Smartphone**, and you can manage WiFi enabled devices such as **WiFiClient Devices** (like a GMC counter) and **WiFiServer Devices.**

  It also allows enabled devices to **submit data wirelessly** to GeigerLog.

- The **I2C Device** support has been greatly expanded, now supporting several dongles and new I2C sensors, in particular for **CO2** measurement with the sensors **SCD30** and **SCD41**.

  The new combinations remain cross-platform compatible.

- A **Manu Device** has been added, which allows to enter data manually, helpful when you can’t get the data electronically, like ruler measured distance of radioactive source from counter, voltage from your DVM, temperature from a home thermometer.

- **Plug and Play** for devices using a USB-To-Serial connection (like GMC counter, GammaScout counter, I2C devices). Their settings can now be auto-detected.

- A **LinFit** (Linear Regression) option has been added to the Graph tools to allow quickly judging the stability of the data.

- More cleanup of the Graphical User Interface

- A lot of refactoring under-the-hood of GeigerLog.

What’s New in GeigerLog 1.2.1 ?

- GeigerLog now supports all current **Gamma-Scout Counters**, including the latest “Online” model, the only one of the breed which allows logging.

- A **Simul Device** has been added, which creates its “counts” with a Poisson random number generator. Helpful for understanding the workings of a Geiger counter and GeigerLog.

- **Search the entire NotePad** for the occurrence of a text like ‘abc’ or ‘123.456’

- The tool **GLpipcheck** was improved

- The Easter Holidays are approaching, and fittingly, an **Easter egg** can now be found in GeigerLog. Do your best – perhaps you will see a dancing **GEIGERA**?

What's New in GeigerLog 1.2 ?

- withdrawn due to a bug with Windows 10

What's New in GeigerLog 1.1 ?

- some major **refactoring** of the code to prepare for future extensions. Most of this will hopefully not be visible to users ;-).
- GeigerLog now supports **HiDPI monitors**
  - HiDPI are High-Resolution monitors with a pixel density higher than FullHD). An example is shown for an 8k (!) monitor. See page 93.
- Editing the GMC counter's **internal configuration**
  - It allows to easily enter SSID, passwords, websites, IDs at your computer and send it to the GMC counter's internal configuration menu with a mouse click. See page 52.
- Support of **MiniMon** devices, primarily CO2 monitors
  - Currently this is supported only for the Linux operating system! See page 68.

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:

  new Calibration Factor = 1 / old Calibration Factor

  Why the change? To make things easier. Read more in chapter Appendix G – Calibration on page 126.

Recommended Reading on the subject from the same author:

All available on the SourceForge site under Articles: **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, and, despite its name, not a Geiger counter, but can measure radioactivity. 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. Die article gives some details on the counter which are not easily available, and shows how to connect GeigerLog to it.

Review of USB-To-I2C Dongles as used by GeigerLog

To connect any I2C device to a desktop or laptop computer requires some hardware, called a dongle. Several were reviewed. The best was the USB-ISS dongle.

[Measuring CO2 with SCD30, SCD41, and MiniMon]

CO_2 measurements with GeigerLog are possible with the I2C devices SCD30 and SCD41, as well as with the MiniMon. All do what they are supposed to do. The SCD41 offers the option of an extended measurement range.

[GeigerLog - Deadtime Correction of Geiger-Counter Events]

The application of a Deadtime-Correction to a Geiger-Counter is reassessed. As a count in a Geiger tube is a so-called "**paralytic**" event, a more complex formula is needed.

(to be published)

[HOWTO - Using Python in a virtual Environment on Linux-v1.0]

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.

[HOWTO - Using PyQt5 and matplotlib on HiDPI monitors (Python3)]

The use of HiDPI monitors is still a challenge. GeigerLog can now run on 8k monitors!

[HOWTO - Read-Write Permissions Serial Port on Linux]

Setting the Read-Write Permissions, relevant to using USB-to-Serial connections.

Author	ullix
Copyright	Copyright 2016, 2017, 2018, 2019, 2020, 2021
License	GPL3, see also Appendix J – License on page 148 You should have received a copy of the GNU General Public License along with GeigerLog. If not, see http://www.gnu.org/licenses/

Table of Contents

Table of Figures

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, barometric-pressure, humidity, CO2, 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, including monitoring of CO2, complemented with a Geiger counter to monitor radioactivity.

GeigerLog itself can be monitored by Smartphone via WiFi.

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

Currently Supported Devices

GMC Devices:

GeigerLog continuous to support the Geiger counters from GQ Electronics [1]) **GMC-3xx**, **GMC-5xx**, and **GMC-6xx** line, including the variants with an additional 2nd Geiger tube, both by wire and wireless connection.

These devices 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 audio-clicks only, like the reviewed **Radiation-v1.1(CAJOE)** [2]) counter.

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)** [3]) semiconductor Geiger counter with a Personal Computer, which has so far not been possible!

1 GQ Electronics LLC, 1001 SW Klickitat Way,Suite 110, Seattle, WA 98134,USA, http://www.gqelectronicsllc.com/

2 GeigerLog-Radiation-v1.1(CAJOE)-Support https://sourceforge.net/projects/geigerlog/files/Articles/GeigerLog-Radiation-v1.1%28CAJOE%29-Support-v1.0.pdf/download

3 GeigerLog-Review Smart Geiger Pro (SGP-001) https://sourceforge.net/projects/geigerlog/files/Articles/GeigerLog-Review%20Smart%20Geiger%20Pro%20%28SGP-001%29-v.1.0.pdf/download

RadMon Devices:

GeigerLog supports the **RadMon+** [4]) hardware, which can provide a Geiger counter as well as an environmental sensor for temperature, barometric-pressure, and humidity.

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

AmbioMon Devices:

GeigerLog supports the **AmbioMon** [5]) hardware, which can provide a Geiger counter as well as an environmental sensor for temperature, barometric-pressure, humidity, and air-quality, driven by an ESP32 microprocessor.

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

Gamma Scout Devices:

GeigerLog fully supports the **Gamma-Scout** [6]) devices ***Standard, Alert, Rechargeable,*** and ***Online,*** including logging with the Online device.

These devices can store recordings in their internal memory. GeigerLog can read this memory.

I2C Devices:

GeigerLog can handle I2C based sensors connected via USB-To-I2C dongles. GeigerLog supports several of them, preferred is the **USB-ISS**. Supported I2C devices: **LM75** (temperature), **BME280** (temperature, barometric pressure, humidity), **TSL2591** (light sensor visible and infrared), **SCD30** (CO2, temperature, humidity), **SCD41** (CO2, temperature, humidity).

More information in recent reviews of dongles and devices [7]) [8]).

LabJack Devices:

GeigerLog supports the **Labjack** [9]) hardware U3 in combination with the ei1050 probe for temperature and humidity.

MiniMon Devices:

GeigerLog supports the **MiniMon** hardware, devices for measuring in-house CO_2 levels using the infrared based NDIR technology.

4 DIYGeigerCounter https://sites.google.com/site/diygeigercounter/

5 This device is in development and not yet publicly available

6 https://www.gamma-scout.com/en/

7 Review of USB-To-I2C Dongles as used by GeigerLog
https://sourceforge.net/projects/geigerlog/files/Articles/Review%20of%20USB-To-I2C%20Dongles%20as%20used%20by%20GeigerLog-v1.0.pdf/download

8 Measuring CO2 with SCD30, SCD41, and MiniMon
https://sourceforge.net/projects/geigerlog/files/Articles/Measuring%20CO2%20with%20SCD30%2C%20SCD41%2C%20and%20MiniMon-v1.0.pdf/download

9 https://labjack.com/

Simul Device Device:

A Simul Device creates synthetic “counts” with a **Poisson random number generator**. Helpful for understanding the workings of a Geiger counter and GeigerLog.

Manu Device:

A Manu Device allows to enter data **manually**, helpful when you can’t get the data electronically, like distance of radioactive source from counter measured with a ruler, but would like to keep the data together and plot them.

WiFiClient Devices:

GeigerLog acts as a **server** to which devices can connect and deliver data. Any of GQ’s WiFi-enabled GMC counters can act as a WiFiClient.

WiFiServer Devices:

GeigerLog acts as a **client** to request data from an external device, which is acting as a server and delivers data when a client, such as GeigerLog, is requesting them.

Main Operations – Logging, Displaying, Analyzing, Monitoring

Logging will be done with a user defined cycle time of 0.1 sec 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 dose rate units of CPM / CPS or µSv/h.
- The **right Y-axis** is reserved for environmental data. If temperature data are shown, the choice of units is between °C and °F.

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

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

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

Analyzing is supported with several **Quality Control** tests, which can be applied to the data. Beyond the standard **statistics** – as a brief summary or 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. **Polynomial Regression** can be applied to test for correlation between the data sets.

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

Remote Monitoring allows to use a Smartphone to monitor the data logged by GeigerLog. This uses the **Smartphone**'s browser; no App is needed.

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 [10]).

It is verified to run with Python 3.6, 3.7, 3.8, 3.9, and 3.10. **Python 3.10** is now considered the version of choice! Earlier versions even of Python 3 are not supported.

In addition to a Python3 environment a few Python modules are needed, which generally are not available in a default installation. Among them is the modern **PyQt5** toolkit.

Step-by-Step installation instructions for Python on **Linux, Windows, Mac,** and **Raspberry** are provided in Appendix H – Installation beginning on page 135.

Installing the software

The software comes in a zipped package containing the Python code and resources like icons and manual. The package is named **`geigerlog-vXYZ.zip`** (xyz is the version number, like 1.3.0).

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’, ‘gres’, ‘gweb’, and ‘gtools’.

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 135.

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 34).

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 108.

10 GeigerLog will NOT run on Python2! The last GeigerLog version running on Python version 2.X is 0.9.06. Use this if you can’t use Python3, but upgrading to Python3 is strongly suggested!

The GeigerLog Window

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

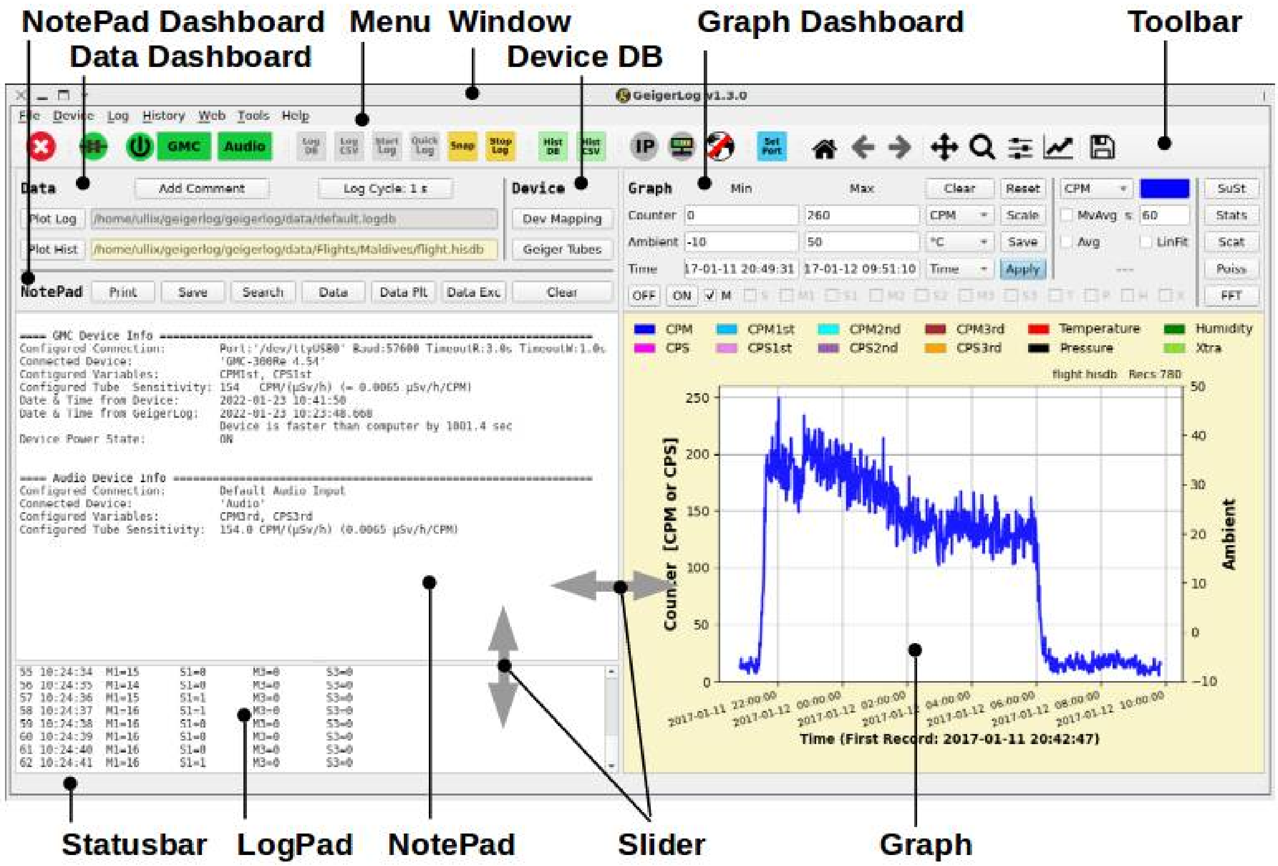


Figure 1: GeigerLog Window with Annotations

Window: The start-up window.

Menu: Menu items; some with keyboard shortcuts in the form of CTRL-X.

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

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

Device Dashboard: Device related settings.

NotePad Dashboard: Handling the NotePad.

Graph Dashboard: Settings to configure your graph.

Statusbar: The bottom line of the window for status and error messages.

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

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

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

Graph: Time-Course Graphs will be shown here.

Quick Tour of GeigerLog

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

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

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

Within 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**. 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 numbers other than 600.

Click the **Clear** button in the **NotePad** Dashboard on the left side, then click the **Data Exc** 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** drop-down button, 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 WiFi (also called WLAN). 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 software connection to the activated devices. Instead of using the menu command you could use the more convenient **Toggle Connection** button in the toolbar, in Figure 2 the second-from-left icon with a plug symbol.

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`.

Figure 2: The Device Toolbar signaling the Connection Status

All possible devices are activated in the configuration file. Top: Before -, Bottom: After – establishing a software connection. The green devices are successfully connected, the red ones failed (here they were not physically connected for demonstration purposes)

By clicking on device icons some info will print into the NotePad similar to this one for a GMC counter and an AudioCounter. Lines in **red** need your attention:

```
==== GMC Device ===========================================================================
Configured Connection:        port:'/dev/geiger' baud:115200 timeoutR:3.0s timeoutW:1.0s
Connected Device:             'GMC-500+Re 1.22'
Configured Variables:         CPM1st, CPS1st, CPM2nd, CPS2nd
Configured Tube#1 Sensitivity:154   CPM/(µSv/h) (= 0.0065 µSv/h/CPM)
Configured Tube#2 Sensitivity:154   CPM/(µSv/h) (= 0.0065 µSv/h/CPM)
Date & Time from Device:      Device clock cannot be read
Device Power State:           ON
Fast Estimate Time:           60 sec. Okay


==== AudioCounter Device ==================================================================
Configured Connection:        Default Audio Input
Connected Device:             'AudioCounter'
Configured Variables:         CPM3rd, CPS3rd
Configured Tube Sensitivity:  154.0 CPM/(µSv/h) (0.0065 µSv/h/CPM)
```

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 up again by clicking the **Dev Mapping** button in the Device Dashboard, or by calling from the menu **Device → Show Device Mappings**.

This printout 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 Temp  Press Humid Xtra
-------------------------------------------------------------------------------------
GMC        : -    -    M      M      M      M      -      -      -     -     -     -
Audio      : -    -    -      -      -      -      M      M      -     -     -     -
RadMon     : M    -    -      -      -      -      -      -      M     M     M     -
Mapping is valid
```

The GMC-Device counter collects Geiger counts at CPM1st, CPS1st, CPM2nd, and CPS2nd, the AudioCounter at CPM3rd and CPS3rd, and the RadMon collects counts at CPM, and temperature, barometric-pressure, and humidity at Temp, Press, and Humid. There are no conflicts.

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

```
==== Device Mappings ===================================================================
The configuration is determined in the configuration file geigerlog.cfg
WARNING: Mapping problem of Variables
Variable CPM                   is mapped to more than one device
Variable Temp                  is mapped to more than one device
Variable Press                 is mapped to more than one device
Variable Humid                 is mapped to more than one device

Device     : CPM  CPS  CPM1st CPS1st CPM2nd CPS2nd CPM3rd CPS3rd Temp  Press Humid Xtra
-------------------------------------------------------------------------------------
GMC        : -    -    M      M      M      M      -      -      -     -     -     -
Audio      : -    -    -      -      -      -      M      M      -     -     -     -
RadMon     : M    -    -      -      -      -      -      -      M     M     M     -
Simul      : M    M    -      -      -      -      -      -      M     M     M     M
Measurements are made on devices from top to bottom, and for each according to configuration.
If double-mapping of variables occurs, then the last measured variable will overwrite the
previous one, almost always resulting in useless data.
Also, Tube Sensitivities will likely be overwritten as well!
```

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 such 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 later chapters 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 until the user specified cycle time has expired before it repeats the process.

This cycle time can be set by clicking the **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 Data Dashboard – Manage Your Recordings on page 20).

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

Before you can log, two things must have been done:

- First, you must have made a connection with at least one device
- Second, you must have loaded a log file

Assuming a connection has been made, we now need a log file. So click the **Log DB** icon in the toolbar to load an existing database 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, or Snap (will be explained shortly). Other functions, which would interrupt logging, like exiting GeigerLog or loading the History from a counter, are also disabled during logging.

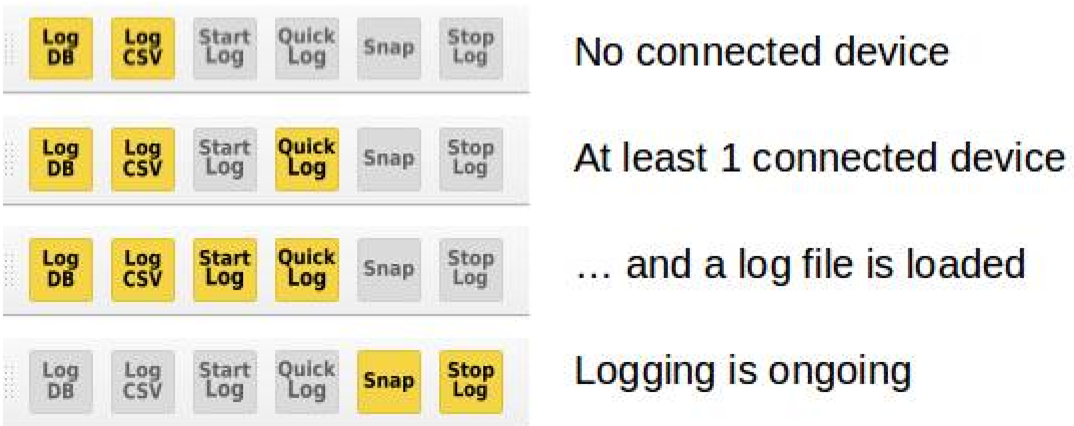


Figure 3: The Logging Toolbar's various stages

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 very 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. Then 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) [11]) 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 Graph Dashboard – Visualize Your Recordings) 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 41.

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!

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

Data Dashboard – Manage Your Recordings

Figure 4: Data Dashboard

The Data Dashboard lets you switch between viewing the Log file and the History file. It has a button to conveniently add a comment to the **active** file, be it Log or History, and you can change the cycle time.

Database 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 active one, currently shown in the graph.

Add Comment

Clicking the **Add Comment** button opens a dialog box allowing you to enter a comment, which will be added to the currently **active** file, which is the one on light-yellow background.

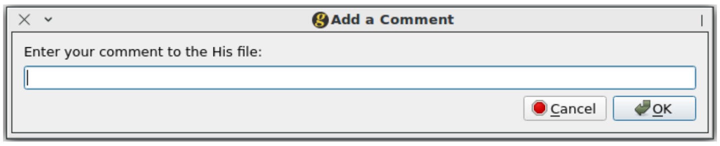


Figure 5: Add Comment to File

This is a convenience function; to enter a comment to the currently not-active file use the Log and History menus.

Log Cycle

The **Log Cycle** button allows to change the logging cycle time.

Clicking the button opens a dialog box where you enter the cycle time in seconds. Any number of at least 0.1 seconds can be entered; numbers less than 0.1 cannot be applied.

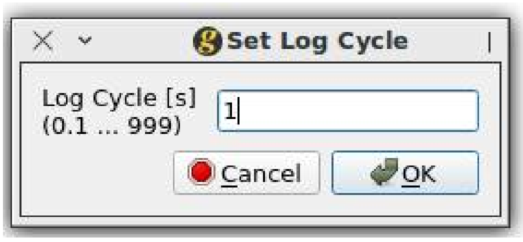


Figure 6: Set Log Cycle

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

Device Dashboard – Manage Device Settings

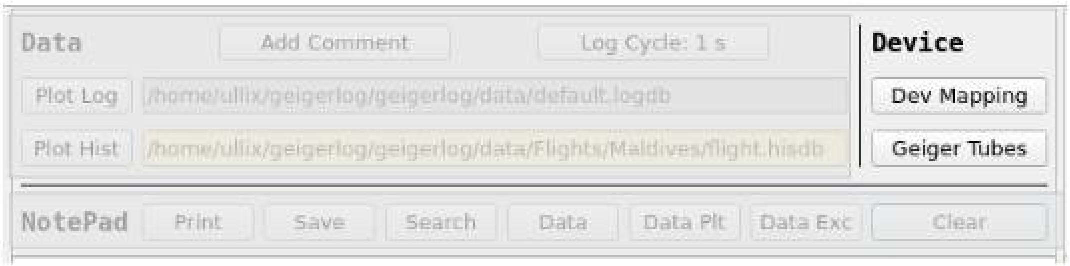


Figure 7: Device Dashboard

Show the Device Mapping and edit the Geiger Tube settings.

Device Mappings

A button to print the Device Mappings into the NotePad; see chapter Device Mappings on page 17.

Geiger Tubes

Clicking the **Geiger Tubes** button opens a dialog box allowing you to view and modify the sensitivities of the Geiger tubes as currently configured. Sensitivities are an important property of your data and impact plotting of graphs. More on this topic in chapter Appendix G – Calibration on page 126.

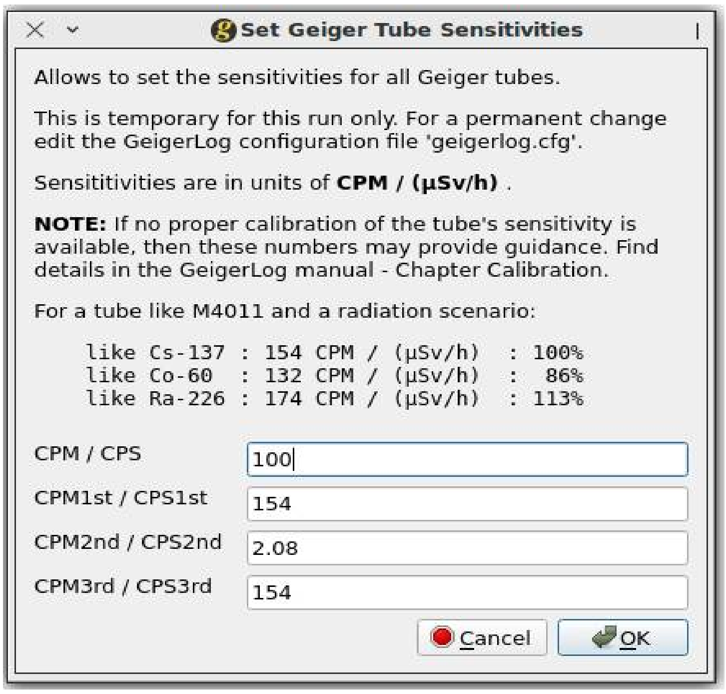


Figure 8: Set Geiger Tube Sensitivities

NotePad Dashboard – Monitor Your Recordings

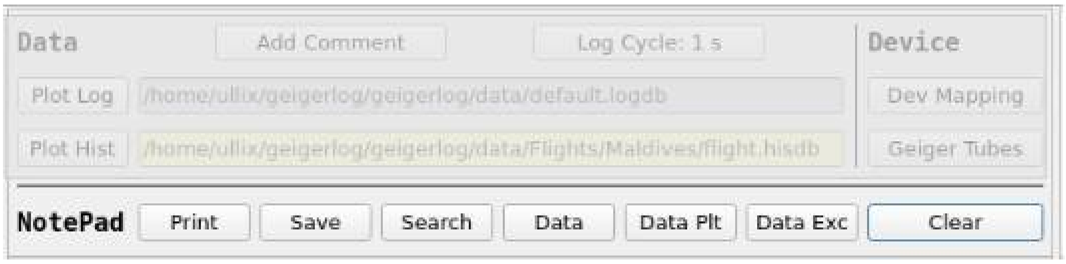


Figure 9: NotePad Dashboard

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

Clear

The **Clear** button clears all content from the NotePad. It may be the most often used button ;-).

Data Exc, Data Plt, and Data

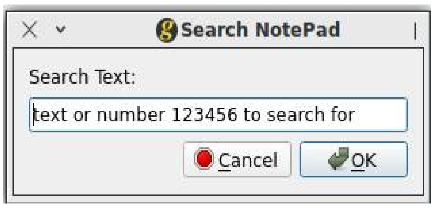
All buttons print data from the **currently active file** – Log file or History file – to the NotePad.

The **Data Excerpt** button prints an excerpt of the data. If the file is short, the whole file is printed, if long, then printing is limited to only a few first and last lines. This may be the second-most-often used button ;-).

The **Data Plt** button prints the data from all data points currently shown in the graph. Helpful to identify individual values.

The **Data** button prints all the data in the file. This button has a second function: when your file is really large, the printing may run for a long time. This button changes into a **STOP** button, allowing you to stop any such long printouts.

Search



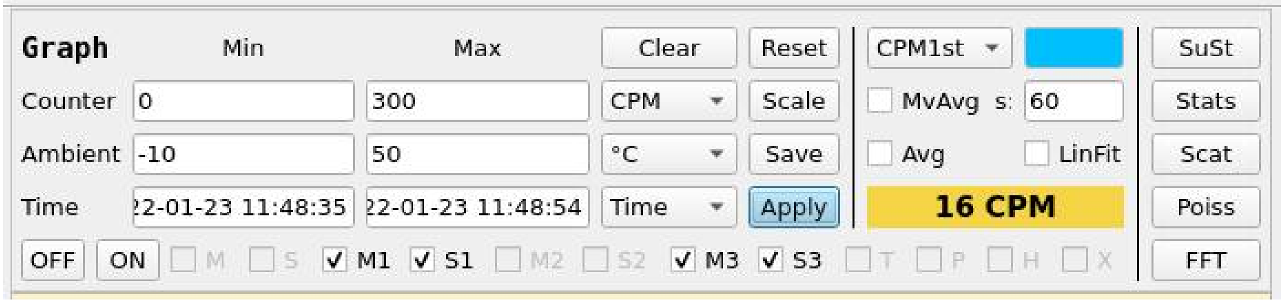
To search the entire NotePad for the occurrence of a text or a number. The button opens a little dialogue box up where you can enter the text to search for. It will search backwards. If it started in the middle of the NotePad and did not find the text, it will restart again from the end.

Also conveniently accessible by pressing **CTRL-F** as is the search function in most editors.

Save and Print

The **Save** button will save the current content of the NotePad to a text file. It will be saved to the data directory and will be named `<filename-of-active file>.notes`. The **Print** button allows to print the NotePad content to paper or to a pdf file.

Graph Dashboard – Visualize Your Recordings



Drawing 1: 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 horizontal X-axis shows the time, and the variables are plotted versus this time on either one of 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 environmental data temperature, barometric-pressure, humidity, and as Xtra anything else, like CO_2, air-quality, light intensity, 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 data file – 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	= 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. CO_2, light, 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. Generally all data are displayed as recorded, but they can be scaled for display, while still being saved unmodified. See chapter ValueScaling and GraphScaling on page 30.

One example is barometric-pressure, which is conveniently displayed as ‘pressure minus 1000’. Since barometric-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.

Another example applies when showing data with an extreme dynamic range, like light-intensity. The data can be compressed e.g. by taking the logarithm, like: log(VAR).

Note: You can also use the options accessible from the Graph Toolbar. Click the icon labeled ‘Edit axis, curve and image parameters’ and select e.g. Scale Log .

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

Save

Clicking the **Save** button will save the current graph to a file in the data directory in the “png” format. The filename is the database file name extended with a DateTime stamp, like: “.../geigerlog/data/default.logdb2022-01-13 12:27:17.png”; it will be printed into NotePad.

Note: Other file saving formats are accessible from the Graph Toolbar. Click the icon labeled ‘Save the Figure’ and select location for saving and the graph format.

Units

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

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

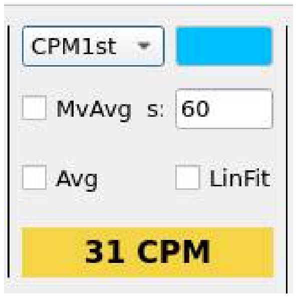
However, when µSv/h is used, the two CPM and CPS based curves will overlap!

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 sensitivity values for each device and tube. This had not always been the case, see discussion in chapter Appendix G – Calibration on page 126.

Right-Y-axis: This Ambient axis is used for all environmental data temperature, barometric-pressure, humidity, and as Xtra anything else, like CO_2, air-quality, light intensity, and other.

The temperature can be displayed in units of either °C or °F.

Selected Variable



The **Selected Variable** segment is a subset of the Graph Dashboard, shown in the picture to the right.

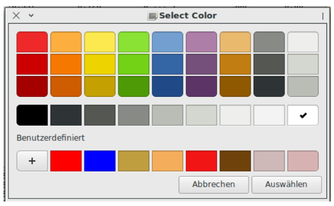
Selected Variable

In this picture **CPM1st** is the **Selected Variable**. By clicking on the drop-down box you can select one of the variables displayed, which thereby becomes the **Selected Variable.** Only variables being currently displayed in the graph can be selected, the others are grayed out in the drop-down box and cannot 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.

The **Selected Variable** is the variable which some of the **Tools** (see below) for analysis and quality control will use in their analysis.

Color Box



This shows the color, in which the **Selected Variable** is drawn in the graph, currently a light blue. By clicking on the Color Box, a dialogue comes up, allowing you to change the color for this variable.

Clicking the Reset button, or reloading the database, or restarting GeigerLog sets the color back to the default color.

MvAvg

If the checkbox **MvAvg** is checked then a **Moving Average** [12]) as a yellow framed line in the color of the Selected Variable will be plotted. The duration of the averaging period is taken from the entry field next to the checkbox. The default duration is 60 sec.

12 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.

With CPS data being recorded once per second, applying a MvAvg of 60 sec to them will basically make a CPM curve out of it. For longer recording times moving averages over 600 or even 6000 may be appropriate.

Avg

If the checkbox **Avg** is checked then a yellow framed horizontal 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 [13]).

LinFit

If the checkbox LinFit is checked then GeigerLog will calculate a **Linear Regression** to the data of the **Selected Variable,** and a straight line representing the regression will be drawn as a yellow framed line in the color of the **Selected Variable**.

If this line is moving up and down, then your data are still fluctuating to much, and you need to collect longer. Once the line remains more or less horizontal (being nearly the same as the average), you may have collected long enough.

Last Value Box

During logging the last value of the **Selected Variable** will be shown in the **Last Value Box** as black letters on a golden colored background, in the picture showing **31 CPM.**

Display Last Values

Clicking on the **Last Value Box** opens the **Display Last Values** window showing the last values of all mapped variables, their device source, and the value they have after Graph-scaling was applied. 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.

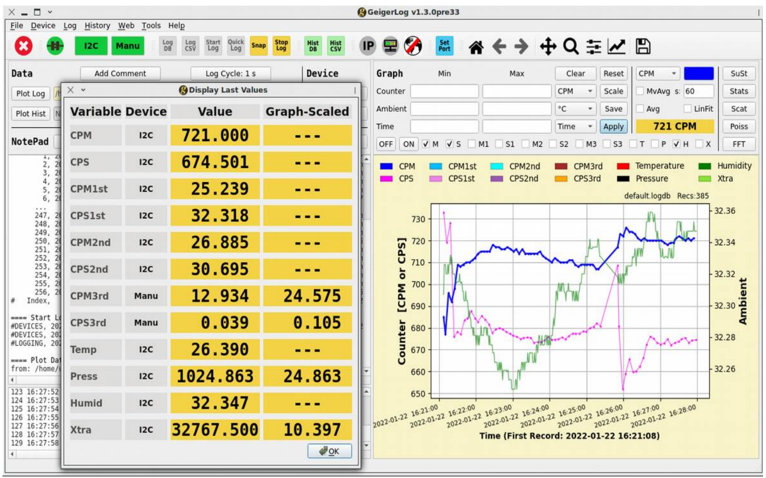
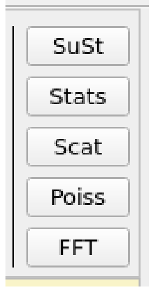


Figure 10: The Display Last Values Pop-up Window

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

Tools



The **SuSt**, **Stats**, **Scat, Poiss,** and **FFT** buttons are tools for the Quality Control of your data.

The **SuSt** button prints a Summary Statistics to the NotePad of all variables currently displayed, the **Stats** button opens a window with more elaborate statistics.

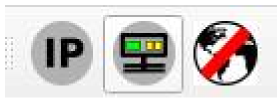
The **Scat** button allows to make a scatterplot of one variable against another one of the configured variables. You can also apply polynomial regressions up to order 7.

The **Poiss** button allows a Poisson test of the data; the **FFT** button does a Fast Fourier Transformation. More details in the chapter Quality Control of your Data on page 37.

The **Poiss** and **FFT** buttons act on the **Selected Variable** only and present their info in a pop-up-window.

Web Enablement

GeigerLog is Web-enabled and can act both as **Client** and as **Server.** It is not only supporting monitoring of GeigerLog by Smartphone, and updating the Radiation World Map, it can also manage WiFi-enabled devices, see: WiFiServer Devices on page 78 and Error: Reference source not found on page Error: Reference source not found.



The functions can be accessed by the menu **Web** or by the toolbar icons as shown on the right. The functions' state is inactive if the icon has a red, diagonal strike-through, and active otherwise.

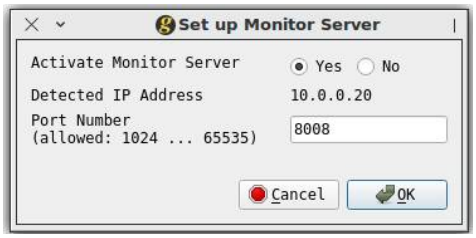
The Web Enablement requires that the computer running GeigerLog is connected to the local network, typically with WiFi access, and that you know its IP address. Then a Smartphone can be used to monitor GeigerLog remotely using this IP address.

IP Address

If you don't already know the IP Address of your computer running GeigerLog, then click the **IP** icon in the toolbar. The pop-up dialogue tells you – in this case – it is '10.0.0.20'. It is a property of the network, and cannot be changed through GeigerLog.

It also provides info on port use of Monitor and GeigerLog's server for any WiFiClient.

Set up Monitor Server



The **Port Number** for the Monitor Server, however, must be set by the GeigerLog user. Default is 8008. Any port in the range 1024 … 65535 can be given. Any lower port, in particular the standard port 80, can NOT be used as this would require administrative privileges and could become a security issue.

When **Yes** is checked at **Activate Monitor Server**, and OK clicked, the Monitor Server can be accessed with any browser by simply entering <IP Address>:<Port Number> – in the case shown it would be: 10.0.0.20:8008 – and you will see the Welcome screen.

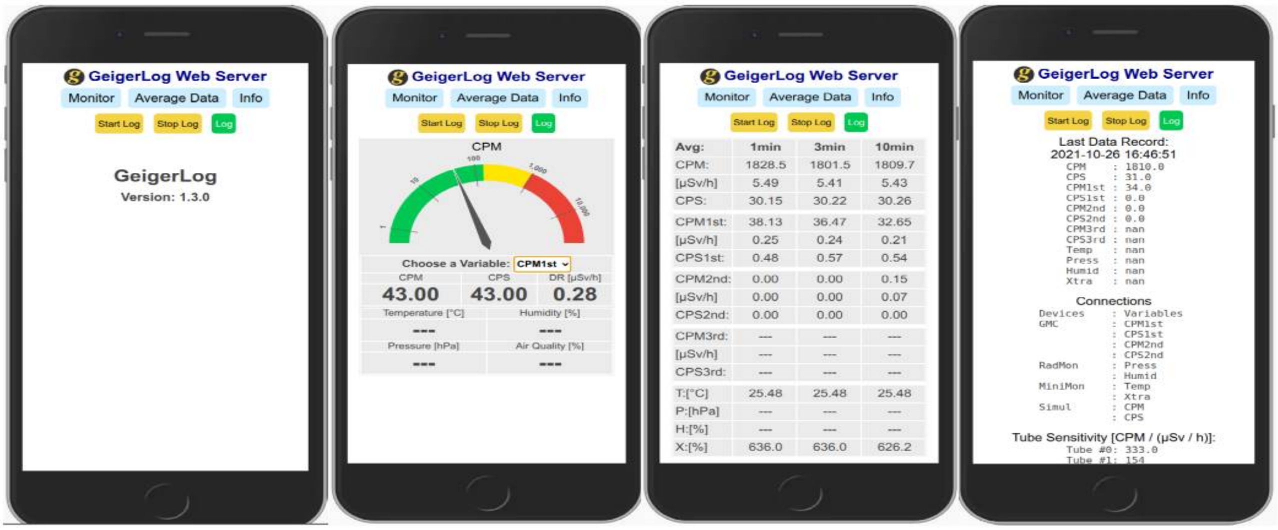


Figure 11: Web Pages delivered by GeigerLog

Tap its menu for other pages.

To auto-activate the Monitor Server at every start of GeigerLog, set `MonServerActivation=Yes` in the GeigerLog configuration file `geigerlog.cfg`.

The speedometer like gauge on the 2^{nd} from left picture shows a green-yellow-red scale, which one is trained to interpret as a ranking from safe to dangerous. But what does actually define, which level is safe, and which dangerous? Find a discussion of the settings I have chosen in chapter On what grounds do we set the radiation safety levels? on page 105.

Set up Radiation World Map

GQ's gmcmap.com site is the only currently supported site. But note that there are some caveats to consider when you use this site, please see Radiation World Maps on page 103!

To use this feature, GeigerLog needs access not only to the local network, but also to the internet.

If you want to contribute to gmcmap.com, you need to register there. This provides you with a UserID and a CounterID. Enter both in the GeigerLog configuration file `geigerlog.cfg` under the heading `[Worldmaps]` into the fields `GMCmapUserID` and `GMCmapCounterID`, resp..

The World Map icon in the toolbar allows to configure the updates. Upon starting GeigerLog the icon is inactive, marked by a red-strike-through. This prevents unwanted updates. Click the icon and a dialog allows you to set up the desired properties of the updater.

Check **Yes** at **Activate Updates** to activate the WorldMap (after the OK button is clicked). Enter the **Update Cycle Time** in minutes. The default is set to 1 hour, but even 12h or 24h do make sense. Governmental sites typically do only DAILY averages of quality controlled data [14])! The values sent to the website by GeigerLog will be averages over exactly this Update Cycle period.

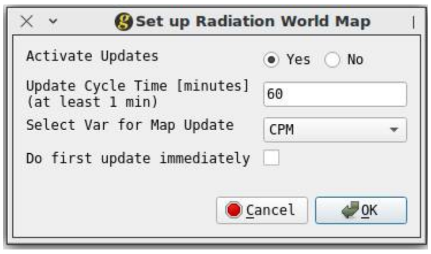


Figure 12: Set up Radiation World Map

Select the **Variable** you want to use for the update. It has to be a CPM* variable, not CPS*!

If you check to **immediately update** (not recommended), the first Radiation World Map Update will be done right after clicking OK, provided that logging is ongoing. If logging is not ongoing, then GeigerLog will wait until it has started, and then do an update. This could result in an update made on a single record only! This is normally undesirable, see Radiation World Maps on page 103! However, it helps getting an immediate feedback on whether the system works correctly. I recommend to have GeigerLog log for a little while at least, and only then activate the Radiation World Map updater!

When GeigerLog does an update, it prints a brief message to the NotePad, like:

```
==== Update Radiation World Map ===========================================
Sending Data   : 2021-11-10 13:37:08  CPM:109.1  ACPM:109.1  uSV:0.71
Server Response: OK
```

To stop updating click the icon and check **No** for active updates.

14 See chapter Radiation World Maps on page 103

ValueScaling and GraphScaling

GeigerLog uses a powerful **Formula Interpreter** to allow scaling of any variable, i.e. modifying the value of a variable by a formula.

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** work on any data loaded from file!

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

The GraphScaling also works on any data loaded from file as soon as they are plotted.

Formula Interpreter

Capitalization has no impact. The default entry is VAL, which means no modification of the value.

Apart from basic math (+ - * /) the formula may include math functions, e.g. x, y could be a value like 123, or itself be a formula:

+ - * /	:	basic math
**	:	raise to the power, e.g. 2**8 (=256)
VAL	:	the value to be scaled, the actual measured value
LOG(x)	:	log to base e; natural log
LOG10(x)	:	log to base 10
LOG2(x)	:	log to base 2
SIN(x)	:	sine
COS(x)	:	cosine
TAN(x)	:	tangent
SQRT(x)	:	square root
CBRT(x)	:	cube root
ABS(x)	:	absolute value
INT(x)	:	integer value
next functions need 2 paramters: x= value, y= deadtime in µs:		
PADEC(x, y)	:	**PA**ralyzing-**DE**adtime **C**orrection
NOPADEC(x, y)	:	**NO**n-**PA**ralyzing-**DE**adtime **C**orrection

Formula example: val – 1000
Formula example: SQRT(val) * 5 + 100
Formula example: ABS((LOG10(val)+1000)/3.14) + 10
Formula example: NOPADEC(val, 120)

Examples:

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

   T_new = T_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_new = SQRT (X_old) (square root)
   X_new = CBRT (X_old) (cube root)

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_new = p_old – 1000

4. The barometric-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. This formula is simple but limited to altitudes of under 50 m [15]). (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_sea = P_alt * (1 + alt / 29.27 * Tv)

5. The barometric-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. This formula is valid at any altitude, but needs T [16]): (sea = sea-level, alt = altitude [m], T = temperature [°C], T chosen as 20°C):

   P_sea = P_alt * (1 - (0.0065 * alt) / (T + 0.0065 * alt + 273.15))(-5.257)**

6. **NOPADEC** (**NO**n-**PA**ralyzing-**DE**adtime **C**orrection). This is a proper deadtime correction for certain devices, but my recent study has shown that this is **NOT** applicable to a Geiger Counter [17]). The formula behind NOPADEC is (deadtime in µs):

   CPM_true = CPM_observed / (1 - CPM_observed * deadtime * 1E-6 / 60)
   CPS_true = CPS_observed / (1 - CPS_observed * deadtime * 1E-6)

7. **PADEC (PAralyzing-DEadtime Correction).** This is needed for use with a Geiger Counter [17]). The formula behind PADEC is:

   CPS_observed = CPS_true * exp(- CPS_true * deadtime)

   This CANNOT be solved analytically! A very fast iterative process is therefore applied in GeigerLog.

---

15 "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 . Hp /29.27 Tv (3.3).*
Hp is the station elevation in meters; and Tv is the mean annual normal value of virtual temperature at the station in kelvins."

16 https://keisan.casio.com/exec/system/1224575267

17 GeigerLog - Deadtime Correction of Geiger-Counter Events. The application of a Deadtime-Correction to a Geiger-Counter is reassessed. As a count in a Geiger tube is a so-called "paralytic" event, a more complex formula is needed. (to be published on the GeigerLog sourceforge site)

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

1. @ Temp: `VAL - 0.23`
2. @ Xtra: `SQRT(VAL)` or `CBRT(VAL)`
3. @ Press: `VAL - 1000`
4. for an altitude of 49 m and Tv = 293 K
   @ Press: `VAL + VAL * 49 / (29.27 * 293)`
5. for an altitude of 85 m and a temperature of 20 °C:
   @ Press: `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 ))`
   or:
   @ CPM: `int(NOPADEC(VAL / 60, 200))`
   @ CPS: `int(NOPADEC(VAL, 200))`
   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!
7. for a dead time of 120µsec (use at any CPM* or CPS*):
   @ CPM: `int(PADEC(VAL / 60, 120))`
   @ CPS: `int(PADEC(VAL, 120)`
   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

To view the current setting of your scaling, click menu **Tools → Scaling Dialog ...** or click the **Scaling** button in the **Graph Dashboard** to see a dialog as in Figure 13.

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

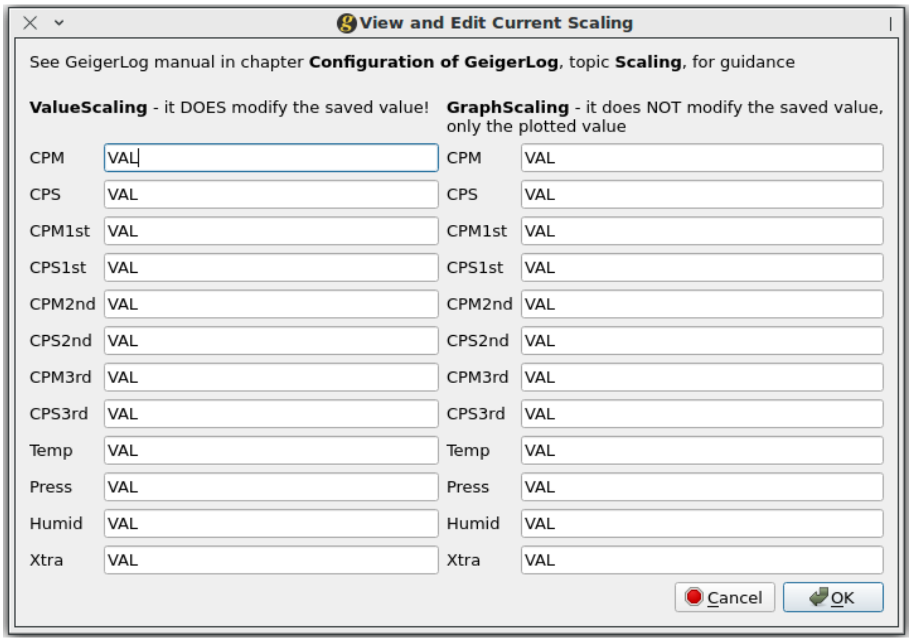


Figure 13: View and Edit Current Scaling

Handling CSV Data-Files

Loading Data from CSV formatted Files

GeigerLog stores its data in SQL-database files. Early versions of GeigerLog had used CSV (**C**omma **S**eparated **V**ariables) text files. Log files had the extension '.log', and History files the extension '.his'. Also, other programs may produce CSV files, which you might want to load into GeigerLog.

This is easily done with functions in the menu: **Log → Get Log from CSV File**, and **History → Get History from CSV File**. You will be offered to load an existing *.log, *.his, *.CSV, *.txt, or *.notes file, which will then be presented in the dialogue. An example is in Figure 14.

Either command creates the exact same database file, except that you cannot add new data to a History file, while you can for a Log file. Such appending to a Log file will most of the time not make much sense, but you do have the option. Generally a History file is the better choice.

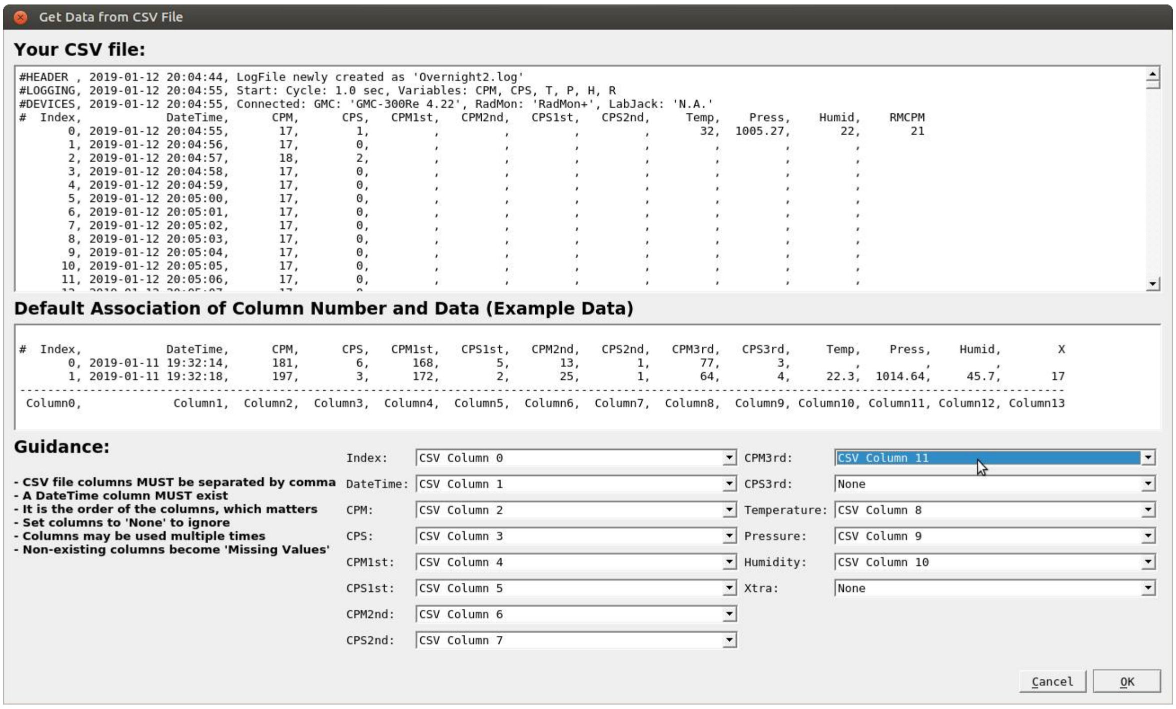


Figure 14: The "Get Data from CSV File" dialogue

The top part shows a segment from the just loaded CSV file, the middle part an example default 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. The data columns can be associated freely with the GeigerLog variables with the exception of the DateTime column. A DateTime column must exist, it must be associated with the GeigerLog variable DateTime, and its preferred format is 'YYYY-MM-DD hh:mm:ss", like "2021-01-23 01:23:45".

Saving Data as CSV formatted Files

Using **Log → Save Log Data into CSV file** and **History → Save History Data into CSV file** will save the respective data as a CSV file with extension 'csv'.

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 15, and GeigerLog exits.

Figure 15: 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 back 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 101). 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 16, giving you detailed info on the problem, and GeigerLog exits.

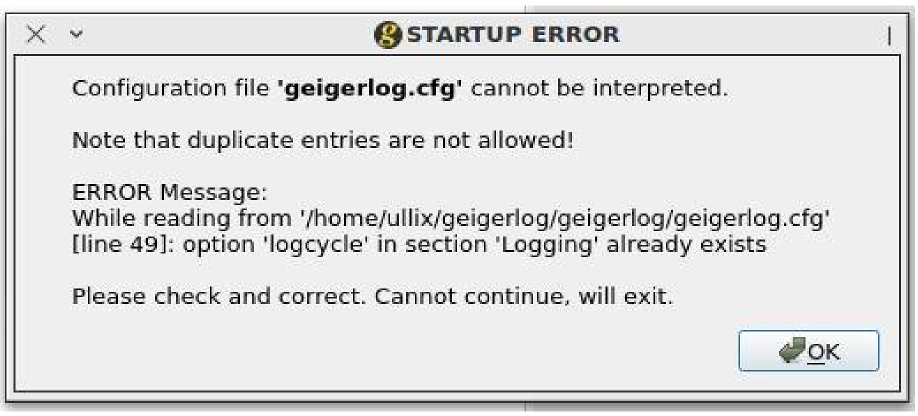


Figure 16: Startup Error on improper definitions in configuration file

Activating devices

Devices can only be worked with if they are activated in the configuration file. If none is activated, then GeigerLog cannot talk to any hardware device, but 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 124.

Conversion between Dose Rate in CPM & CPS and Dose Rate in µSv/h

This requires that a calibration exists, specific for each counter and its tube. This is not a trivial task; read more on this in chapter Appendix G – Calibration.

GeigerLog uses the tube's **Sensitivity** for this. It is given in units of **CPM / (µSv/h)**. The sensitivity value tells you how many counts-per-minute a tube will generate in a gamma-radiating environment of 1 µSv/h. The more sensitive a tube is, the higher the sensitivity values will be!

Some sensitivity examples for tubes:

154	CPM / (µSv/h)	: M4011	– a tube used in many GMC and various low-cost counters
348	CPM / (µSv/h)	: LND7317	– a very sensitive (and expensive) Pancake tube
2.08	CPM / (µSv/h)	: SI3BG	– an insensitive tube, used as 2nd tube in the GMC500+

The conversion between dose rates in CPM and in µSv/h is done with these formulas:

Dose Rate [CPM] = Dose Rate [µSv/h] * Sensitivity [CPM / (µSv/h)]
Dose Rate [µSv/h] = Dose Rate [CPM] / Sensitivity [CPM / (µSv/h)]

Note: This use of 'Sensitivity' constitutes a difference between GeigerLog Version 1.0 and later and earlier versions, which used 'Calibration Factor' for the conversion instead! But the difference is simple: Sensitivity is the inverse of the Calibration Factor.

The reason for the change is that the quantity **Sensitivity** is easier to grasp, and is the standard by which established tube manufacturers report this property of their tubes.

Some Issues with the dose rate conversions

This sensitivity 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, simply due to the design and material used for the case! Furthermore, it also depends on the type of radioactive environment! An in-depth discussion of some of the issues with calibration is found in chapter Appendix G – Calibration on page 126.

Sensitivities should be provided by the manufacturer of a Geiger counter, but may not be stated explicitly in a spec sheet, instead only by embedding them in the firmware of the device. And a user may or may not be able to read them out of the firmware.

GeigerLog uses the published sensitivities and applies them when the configuration files specifies:

```
sensitivity = auto
```

To use anything else, replace `auto` with your numeric value for sensitivity.

For the GMC Geiger counters the sensitivity is built into the firmware [18]) and can be read out by GeigerLog. However, it recently turned out that one of these factors was wrong by a factor of ~2.5 [19]). GeigerLog uses the corrected value, but, if needed, you can always set the original, (wrong) value [20]).

To quickly see the current setting of sensitivities use menu **Device → Geiger Tubes …**, or press the **Geiger Tubes** button in the **Device Dashboard** for a dialog as in Figure 17.

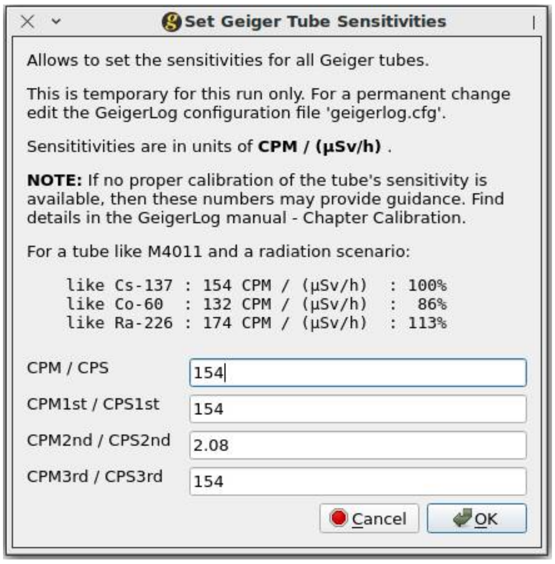


Figure 17: Set Geiger Tube Sensitivities

Four different settings can be seen and edited and will be used for counter data recorded to the variables CPM / CPS, CPM1st / CPS1st, CPM2nd / CPS2nd, CPM3rd / CPS3rd, resp.

This dialog provides a convenience function for experimentation with different tubes. It allows to redefine the sensitivity currently in use. This redefinition can be applied at any time, and its changes will be effective immediately, but it is discarded when the program ends, or a device re-activation occurs.

For lasting changes edit the configuration file `geigerlog.cfg`.

18 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 126.

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

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

Quality Control of your Data

The **SuSt**, **Stats, Scat**, **Poiss**, and **FFT** buttons in the Graph Dashboard help you to check the quality of your data. In addition menu option **Tools** → **Show Plot Data** provides further help.

These functions do:

- **SuSt** gives a brief **Summary Statistics**
- **Stats** a more in-depth **Statistics**
- **Scat** shows a **Scatterplot** of one variable against any other with a Polynomial Regression
- **Poiss** a test for the data having **Poisson Distribution** properties
- **FFT** provides a **Fast-Fourier-Transform**
- **Show Plot Data** prints data to the NotePad, but only those shown in the plot

The first four functions use only the data and variables currently shown in the plot for their calculations! If you want to see the result for all the data in the file, click the **Reset** button in the Graph Dashboard first.

Poiss and **FFT** can only work with one variable at a time. Select this variable with the **Selected Variable** drop down box, see Graph Dashboard – Visualize Your Recordings on page 23.

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

SuSt – Summary Statistics

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

```
==== Summary Statistics of Variables selected in Plot ========================
File       = /home/ullix/geigerlog/geigerlog/data/default.logdb
Filesize   =  1,303,552 Bytes
Records    =     37,998 total,       31,787 shown in Plot
Time Span  =    0.69123 d total,    0.36820 d shown in Plot
Avg. Cycle =        1.6 s total,        1.0 s shown in Plot
          [Unit]   Avg ±StdDev   Variance       Range         Recs  Last Value
CPM     : [CPM]  15.91 ±3.8         14.46       6 ... 31      31787        N.A.
CPS     : [CPS]   0.27 ±0.514        0.26       0 ... 4       31787        N.A.
CPM3rd  : [CPM]  14.98 ±3.93        15.47       2 ... 32      15860        N.A.
CPS3rd  : [CPS]   0.50 ±0.712        0.51       0 ... 6       15889        N.A.
```

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

The reason for this lies in the properties of a Poisson Distribution, which is the relevant statistics for radioactive events. For an introduction to Poisson Distribution and its statistics see my "Potty Training for Your Geiger Counter" article available on SourceForge in the Articles folder [21]).

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

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

21 https://sourceforge.net/projects/geigerlog/files/Articles/

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,    RMCPM
    0, 2018-08-19 13:41:50,       16,         1, 27.2, 1012.52,    36,       14
...
 2388, 2018-08-19 15:41:14,  6778.79,    121.9,
 2389, 2018-08-19 15:41:17,  6761.01,   116.66,
 2390, 2018-08-19 15:41:20,  6773.56,   110.38,
 2391, 2018-08-19 15:41:23,  6794.47,   109.34,
 2392, 2018-08-19 15:41:26,  6803.89,    96.84,
 2393, 2018-08-19 15:41:29,   6801.8,   127.15, 29.2, 1011.5,     35,       22
 2394, 2018-08-19 15:41:32,  6796.57,   114.57,
```

Poiss – Histogram with Poisson Fit

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

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

The next two figures provide examples of histograms with a Poisson fit; Figure 18 for low count rates as in a background measurement, and Figure 19 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.

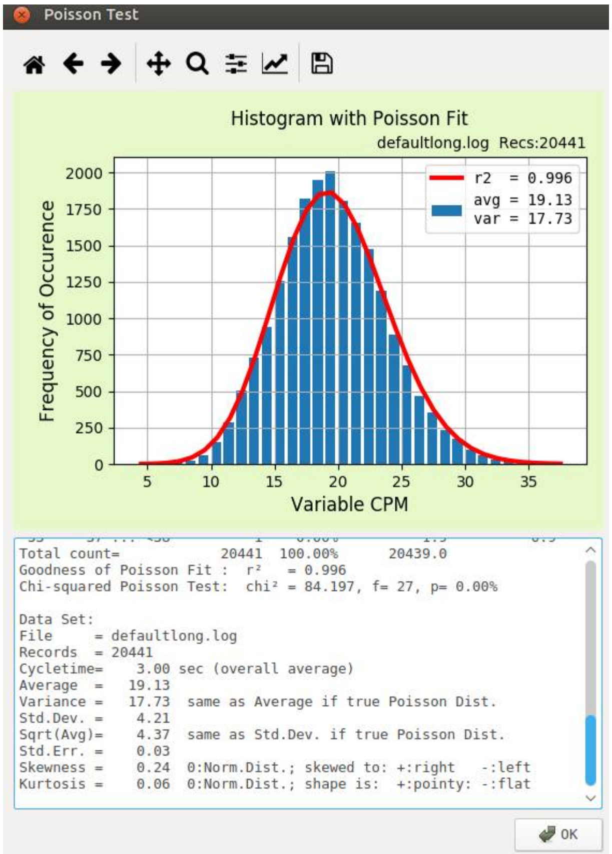


Figure 18: Histogram of Low Count Rate

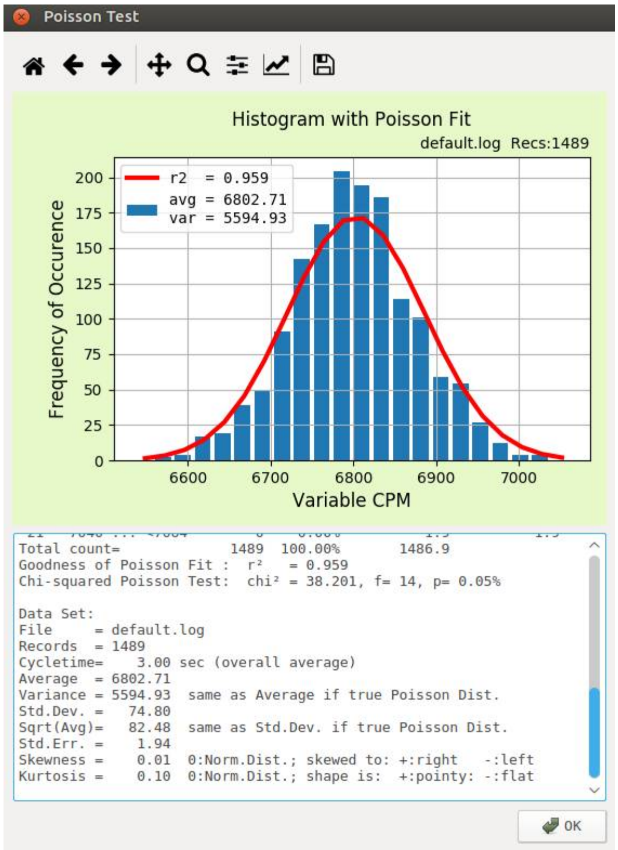


Figure 19: Histogram of High Count Rate

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 20. The data were recorded by logging in the CPM mode.

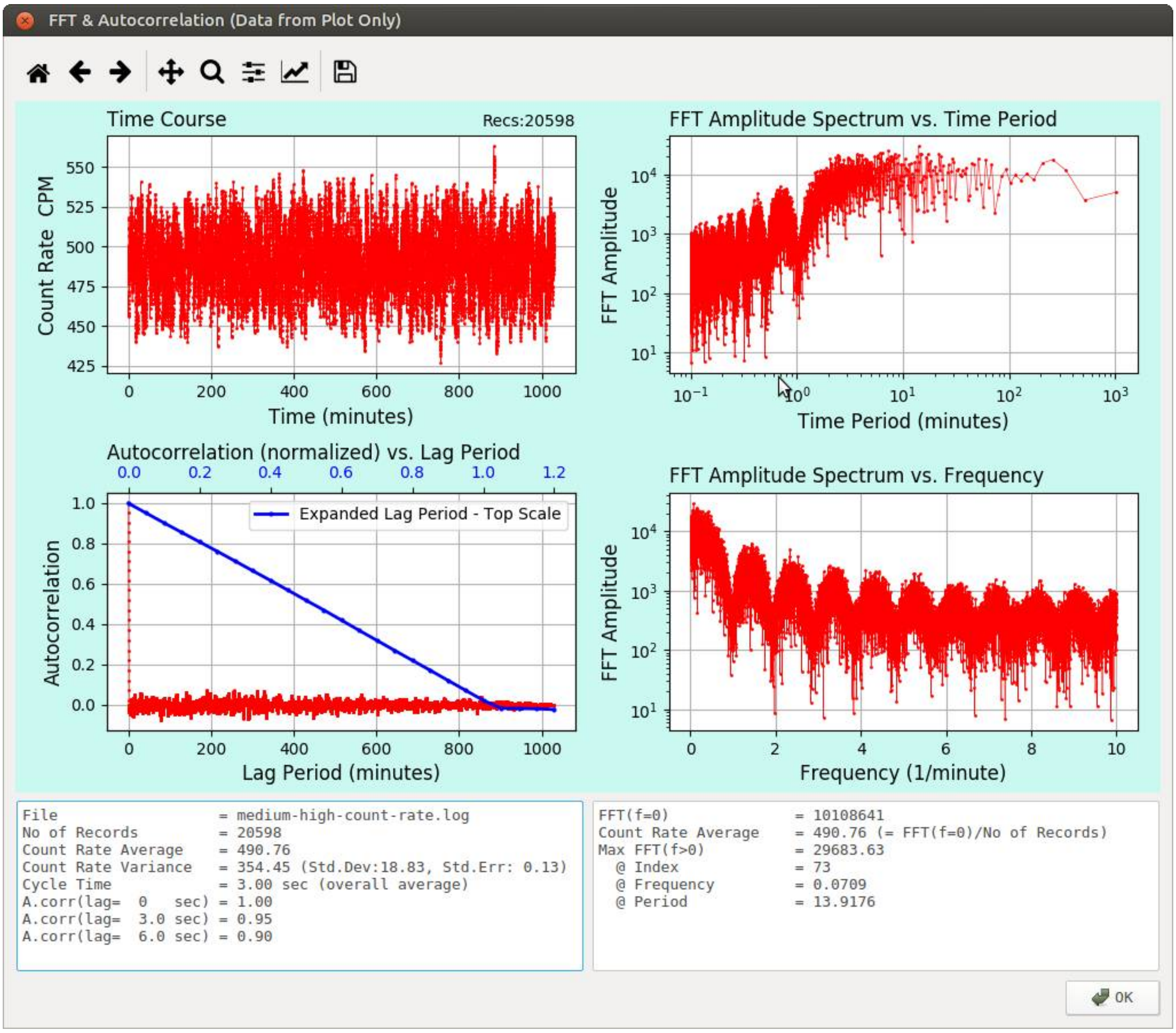


Figure 20: FFT Analysis of Medium Count Rate Measurement

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

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

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

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

Such a relationship can be quantified by calculating the autocorrelation of a signal. This is shown in the bottom left diagram of Figure 20. 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.

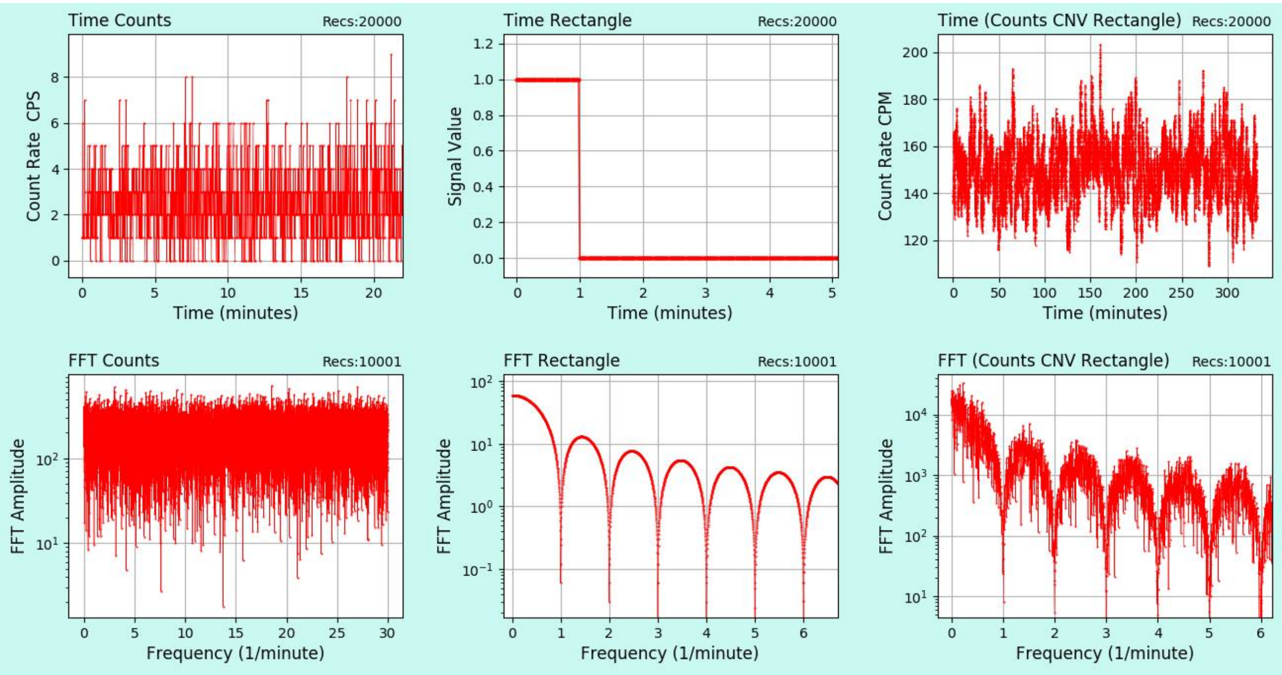


Figure 21: Demonstration of the impact of convolution on an FFT spectrum

(Synthetic data, CPS=2.5)

This can be nicely demonstrated using synthetic recording. In Figure 21 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.

Scat – 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 barometric-pressure has an influence on the temperature?

More relevant to Geiger counters was the recent observation [22]) that a GMC Geiger counter equipped with a M4011 tube (black version) was significantly sensitive to temperature, which it shouldn't be!

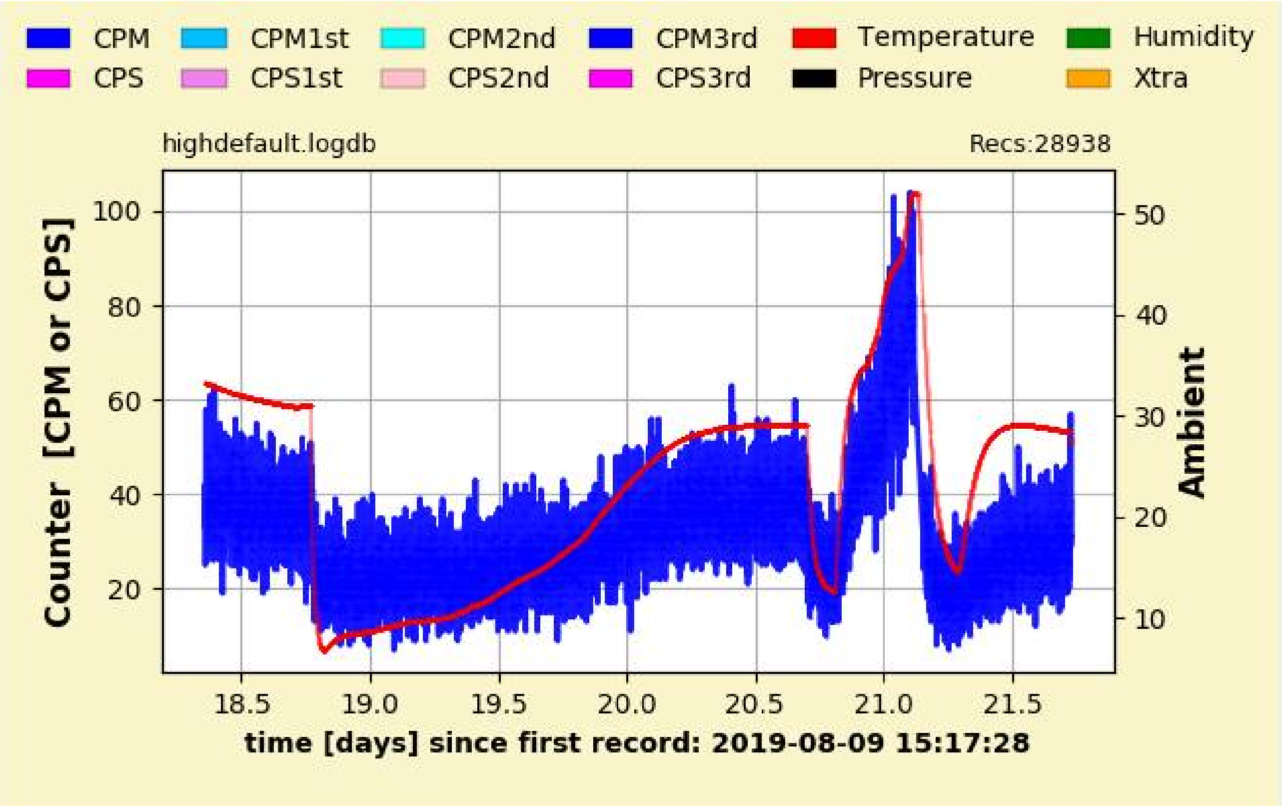


Figure 22: CPM and Temperature in a Temperature Experiment with a Geiger Counter

(GMC-300E+ with a M4011 tube - black version)

It is obvious from the time course plot in Figure 22 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.

Press the Scat button. 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 23.

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.

22 http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=7475

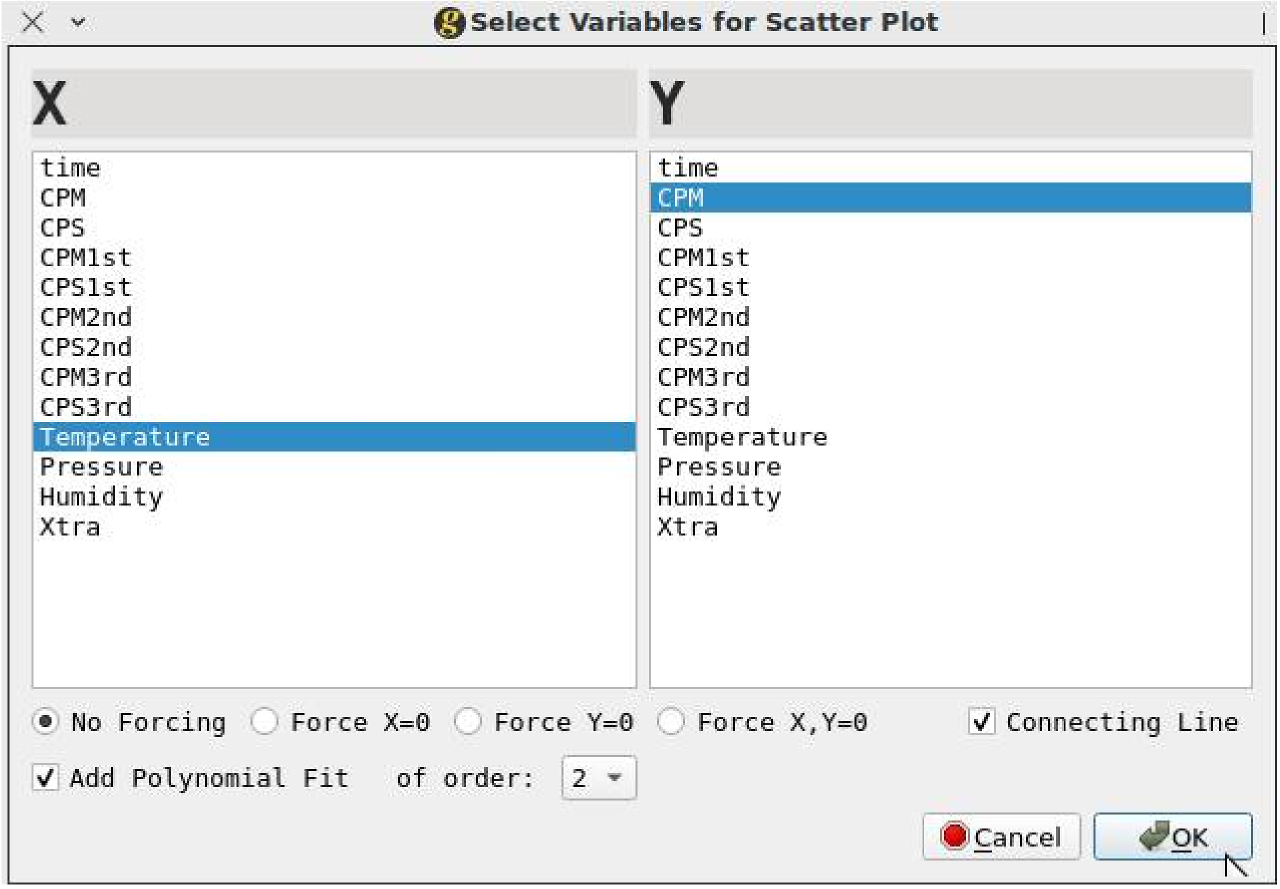


Figure 23: Select Variables for Scatter Plot

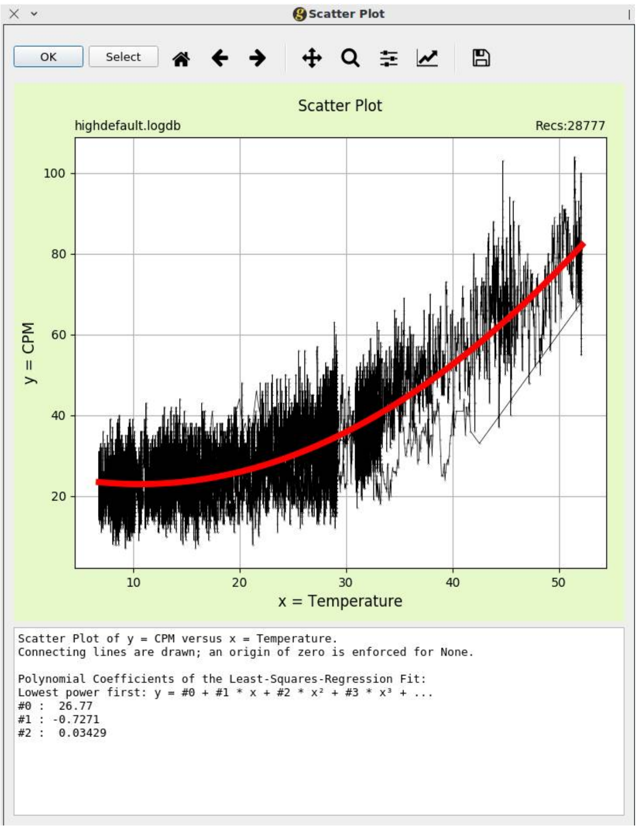


Figure 24: Scatter Plot with 2nd Order Polynomial Fit

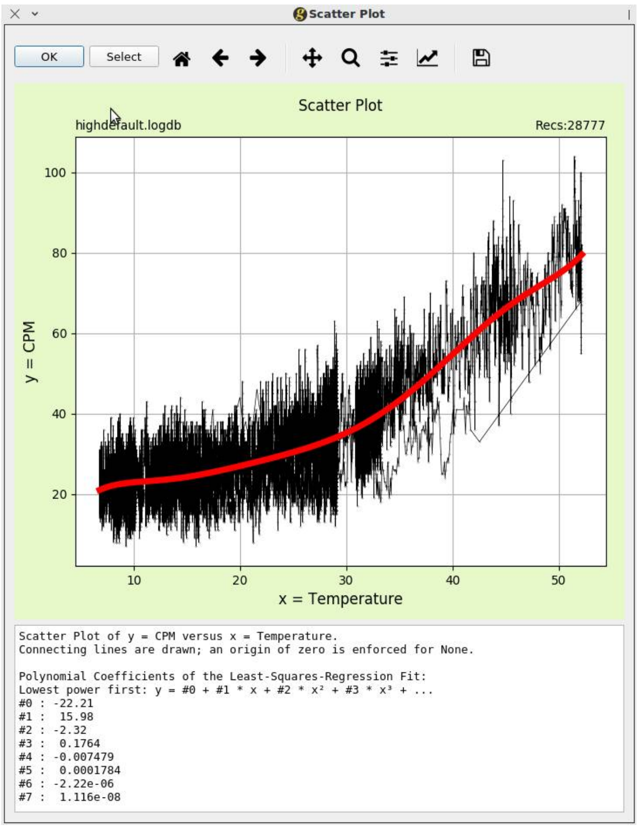


Figure 25: Scatter Plot with 7th Order Polynomial Fit

In Figure 24 you see a scatter plot of y = CPM versus x = 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 23. The choices you had just made, are maintained. Let's change the order of the fit from 2 to 7, and click OK.

Figure 25 shows the result with now a fit of the same data with a polynomial of 7^{th} 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^{nd} 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 55.

In the Select Variables dialog (Figure 23) choose the variables as X = CPM (digital data) and Y = CPM3rd (audio data). Uncheck **Connecting Line**, and choose a fit of order = 1, i.e. a linear fit. The result is shown in Figure 26, demonstrating the excellent correlation between the two.

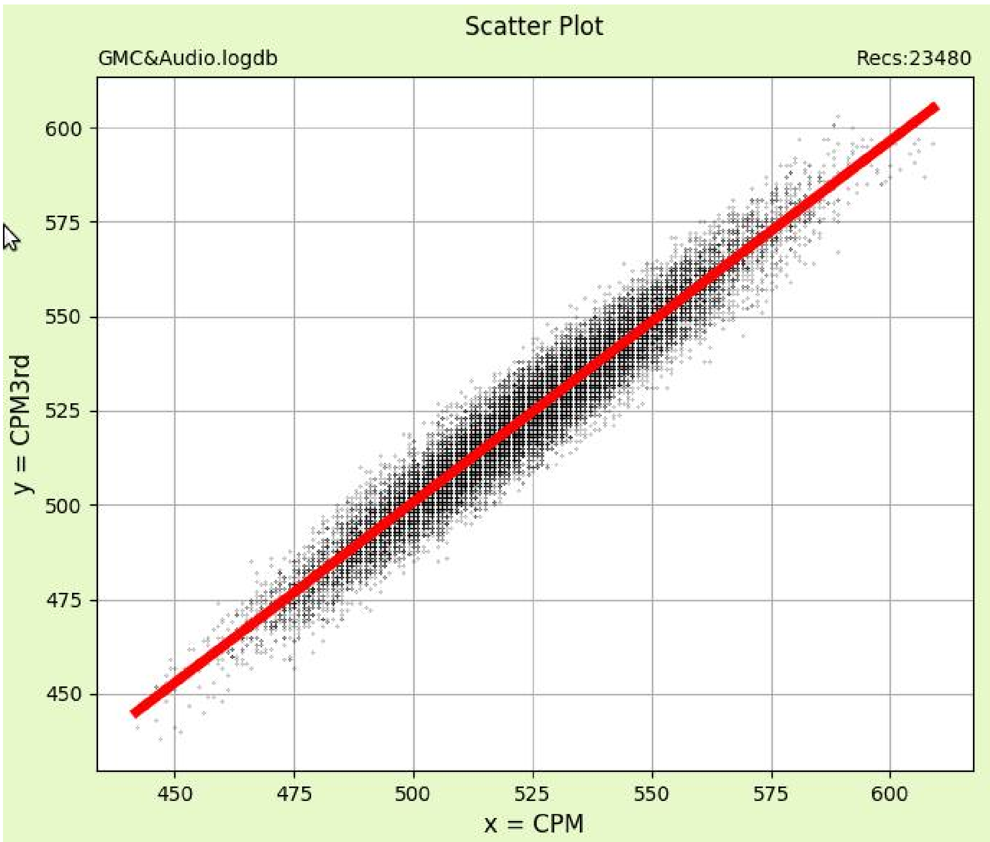


Figure 26: Scatter Plot of CPM3rd=Audio versus CPM=Digital Signal (Using a GMC-300E+ Geiger Counter)

Show Plot Data

Sometimes you want the numerical values of your variables. In the **NotePad Dashboard** click the **Data Plt** button, or use menu **Tools→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 27. Makes it easy to inspect values within a limited range.

Print: You can print this selection on paper or as a pdf file with the **Print** button in the **NotePad Dashboard** or via menu **File → Print NotePad.**

Save: You can also save this printout to a file using the **Save** button in the **NotePad Dashboard** or menu **File → Save NotePad to File.** This will create a CSV (Comma Separated Values) file. It can be read back into any GeigerLog, which is a convenient way to forward select data to other users.

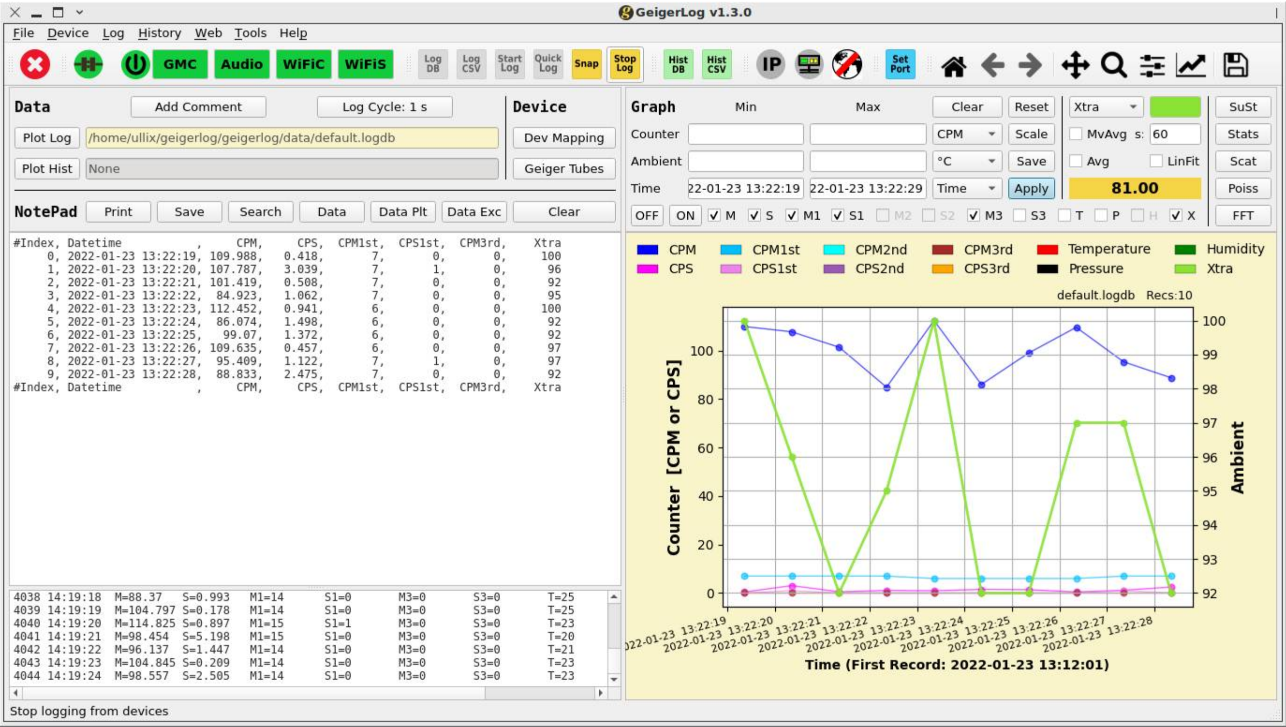


Figure 27: Show Plot Data showing only the values of data currently in the plot

Device-specific Considerations

These devices are available:

1. GMC Devices
2. AudioCounter Devices
3. RadMon Devices
4. Gamma-Scout Devices
5. I2C Devices
6. LabJack Devices
7. MiniMon Devices - CO2 Monitor
8. Simul Device
9. Manu Devices
10. WiFiServer Devices
11. Error: Reference source not found

GMC Devices

GMC Devices are being fully controlled by GeigerLog. All communication with the device is error-checked and corrected if possible.

Device functions like Speaker, Alarm, Saving Mode, Date&Time, Calibration, Threshold, can be read by GeigerLog, and some can also be set, which is much more convenient than clicking through the counter's menus!

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 (except that you cannot add more data to it).

Connecting

GeigerLog, beginning with its version 1.3.0, can auto-detect the presence of a GMC counter and make the correct settings, thus providing for **plug-and-play use of GMC counters**!

However, this is possible only as long as the manufacturer GQ maintains the use of hardware and software in the counter as it currently is [23]). If this changes, then you need to do these settings manually. Also, some older GQ counters use different chips; while they do work with GeigerLog, you'll have to make the serial connection settings manually.

For that, first see chapter Appendix B – Connecting Device and Computer using a Serial Connection on page 109 for an important explanation!

After starting GeigerLog select menu **Device → Connect Device** as explained in Establishing a GeigerLog Connection to Your Devices on page 16.

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

If successful, you'll see the GMC icon turn green, and a confirming message is printed to the NotePad. You're set to go.

23 Current GMC counter all use the USB-To-Serial chip **CH340** (VID:PID=0x1a86:0x7523) from WCH. http://wch-ic.com/products/CH340.html

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 124.

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 → Set GMC Configuration …**). GeigerLog's GMC device power icon will change its state from Power OFF (red) to Power ON (green). The icon is gray when the device is not connected and thus GeigerLog could not 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 [24])!

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:

24 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.

CPM	: Counts Per Minute as the sum of both tubes (makes no sense [25]))
CPS	: Counts Per Second as the sum of both tubes (makes no sense)
CPM1st	: Counts Per Minute for the 1st tube, the standard tube
CPM2nd	: Counts Per Minute for the 2nd tube, the low-sensitivity tube
CPS1st	: Counts Per Second for the 1st tube, the standard tube
CPS2nd	: Counts Per Second for the 2nd tube, the low-sensitivity tube

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

CPM = CPM1st = CPM2nd
CPS = CPS1st = CPS2nd

An example of a Logging with a GMC-device in combination with an AudioCounter as a 2nd device is shown in Figure 29 in chapter AudioCounter Devices.

History Background

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 neither 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!

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

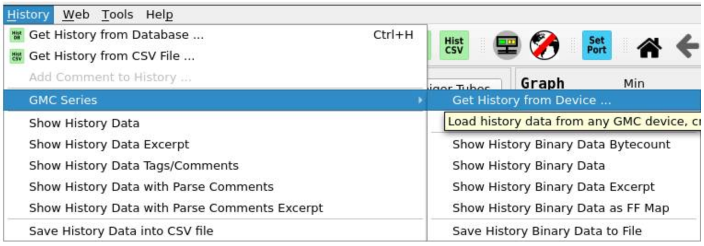
If you don't need the content of the memory, I suggest to erase it every once in a while. Unfortunately, on the older counters this can be done only with a Factory Reset. Some newer counters provide a separate command to erase the memory.

> **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`!

History Handling

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

The button **Hist DB** loads data from a database file created by a previous history download, see more at Menu – History. The button **Hist CSV** reads a file in CSV format and creates a database.



When a GMC device is activated then the menu History offers additional options. Once a GMC device is also connected, you can **download** the History from the device.

Note that with ongoing Logging this History Downloading cannot be done, as the GMC Devices are limited computationally and stumble with parallel logging and downloading!

Additional commands in the menu History are:

History → GMC Series → Get History from GMC 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. Such a History example – data collected during a long distance flight – is shown in Figure 1.

Other commands provide details about the binary data and data structure.

History → GMC Series → Save History Binary Data to File save the raw data to a file with extension 'bin'.

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 to Mini-B Cable is needed like this:
https://www.amazon.co.uk/AmazonBasics-Male-Mini-B-Cable-Feet/dp/B00NH11N5A

GeigerLog's 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.

Editing the GMC counter's internal Configuration

From the menu use **Device → GMC Series → Set GMC Configuration …** to bring up the dialogue box shown in Figure 28.

The upper half applies to all GMC counters and allows to change Power, Alarm, Speaker, History Saving Mode, and Calibration Points. The calibration points are the internal ones of the counter. GeigerLog uses its own calibration.

The lower half is for WiFi-enabled counters only, which are the GMC-320V5, and the GMC-500/600 series counters. For all other counters this lower part is grayed out and cannot be edited.

If you have already set the WiFi related information in the GeigerLog configuration file `geigerlog.cfg`, then you fill out this form by just pressing the **Show User Configuration** button. Pressing the **Show Counter's Active Configuration** button will reset all information to what just had been read out from the counter.

Upon pressing **OK** everything will be written into the counters configuration memory.

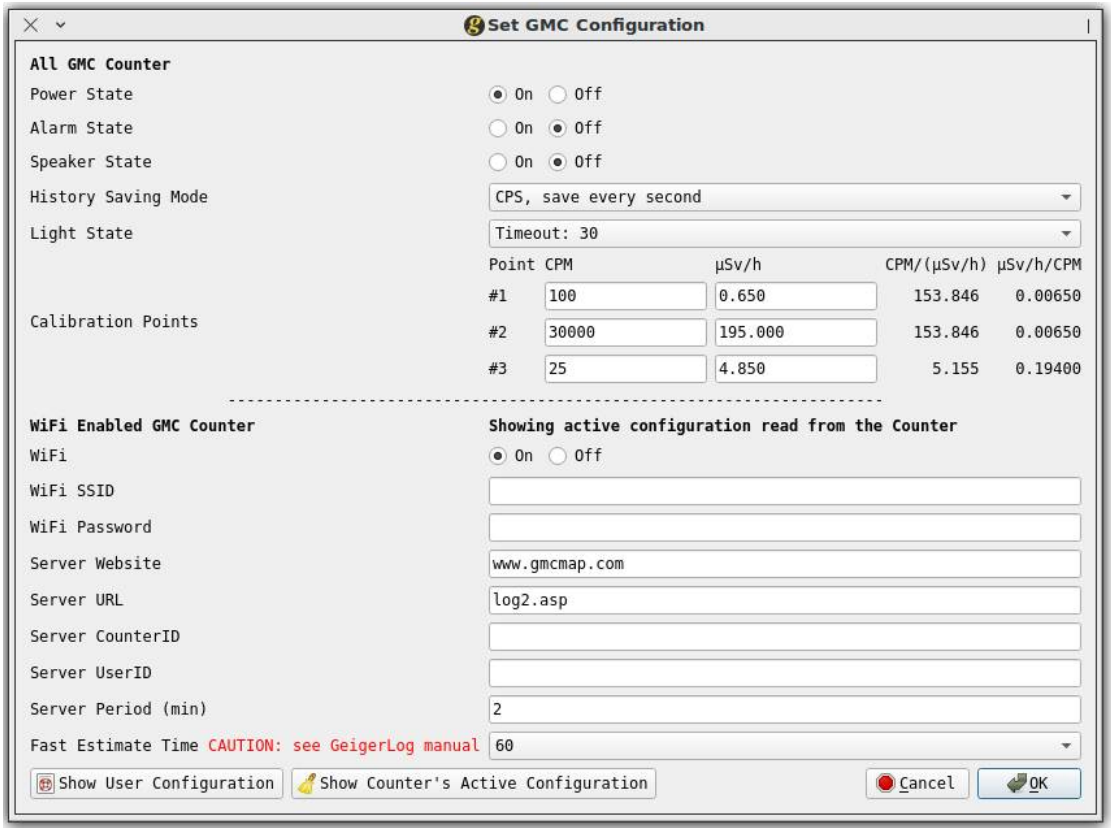


Figure 28: The dialogue to edit the GMC counter's internal configuration. This screenshot is from a GMC-500+ counter, which has 2 tubes.

Fast Estimate Time (FET) CAUTION !

The last option in the WiFi section is the **Fast Estimate Time** (FET). While this is not related to WiFi, it exists only with some WiFi enabled counters, like GMC-500 and 600 with newer firmware.

By default this is set to 3 seconds, which by GQ is labeled “Dynamic” mode. It was found to be the reason for completely distorted measurements, even for the creation of counts where there are none! It is discussed in depth in this forum topic: http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=9497, and also: http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=9506 .

It is set to 3 (aka “Dynamic”), 5, 10, 15, 130, 30, 60 seconds. The algorithm is not fully disclosed by GQ, except that at 5 seconds and greater the CPM value is estimated on the basis of counts for number of seconds selected by the simple formula:

$$CPM_{FET} = \text{Sum of CPS counts during FET seconds} * (60 / FET)$$

This necessarily results in distorted Poisson distributions, and in the default Dynamic setting even in false data. A setting of FET = 60 obviously switches FET off.

It is strongly suggested to always switch the FET setting OFF, i.e. set it to 60 seconds!

Unfortunately, after every Factory Reset this will be back to ‘3’, as this is the default, and will have to be switched off again.

WiFi Equipped GMC Counter

Out of GQ's Geiger counter some are WiFi equipped: series GMC-320+V5, GMC-500, GMC-600. Unfortunately, the intended use of this WiFi is only for the update of their Radiation World Map on their gmcmap.com server (see Radiation World Maps on page 103, and Set up Radiation World Map on page 29).

However, with a bit of hacking the data can now be redirected to GeigerLog, see Error: Reference source not found on page Error: Reference source not found. Here is shown the normal operation for updating the gmcmap server.

After registration you will have received a User ID (AID) and a counter ID (GID). GMC counters are limited to 3 variables named: CPM, ACPM, uSV [26]).

The map CPM is the instantaneous value of the counter's CPM. The ACPM is some averaged value of CPM, but the period over which the averaging occurs is unknown. The uSV is calculated from CPM as CPM / 154 CPM / (µSv/h), i.e. from the instantaneous value, and NOT from the more meaningful average!

Their URL, when contacting the gmcmap server, looks like:

http://gmcmap.com/log2.asp?AID=123&GID=456&CPM=41&ACPM=41.18&uSV=0.27

The gmcmap.com server will answer with:
\r\n<!--   sendmail.asp-→\r\n\r\nOK.ERR0

or, when AID and/or GID is missing or incorrect:
User is not found.ERR1.
Counter is not found.ERR2.

The counter wants to see the answer, but only the ERR0, ERR1, ERR2 portion seem to matter.

26 https://www.gmcmap.com/AutomaticallySubmitData.asp

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 fully supported by GeigerLog. How to operate, connect, configure and run such devices is described in the article **GeigerLog-AudioCounter-Support** [27]).

An example for a low-cost, audio-only counter is the **Radiation-v1.1(CAJOE)** counter; see the review [28]).

But even some very advanced counter designs offer only an audio output, like in particular this also reviewed, semiconductor based radioactivity detector **Smart Geiger Pro (SGP-001)** [29]).

The article also contains a comparison of the digital measurement with the audio measurement using a Geiger counter, which can do both simultaneously, here a GMC-300E+. The results fully confirm the validity of the two different methods. Figure 29 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.

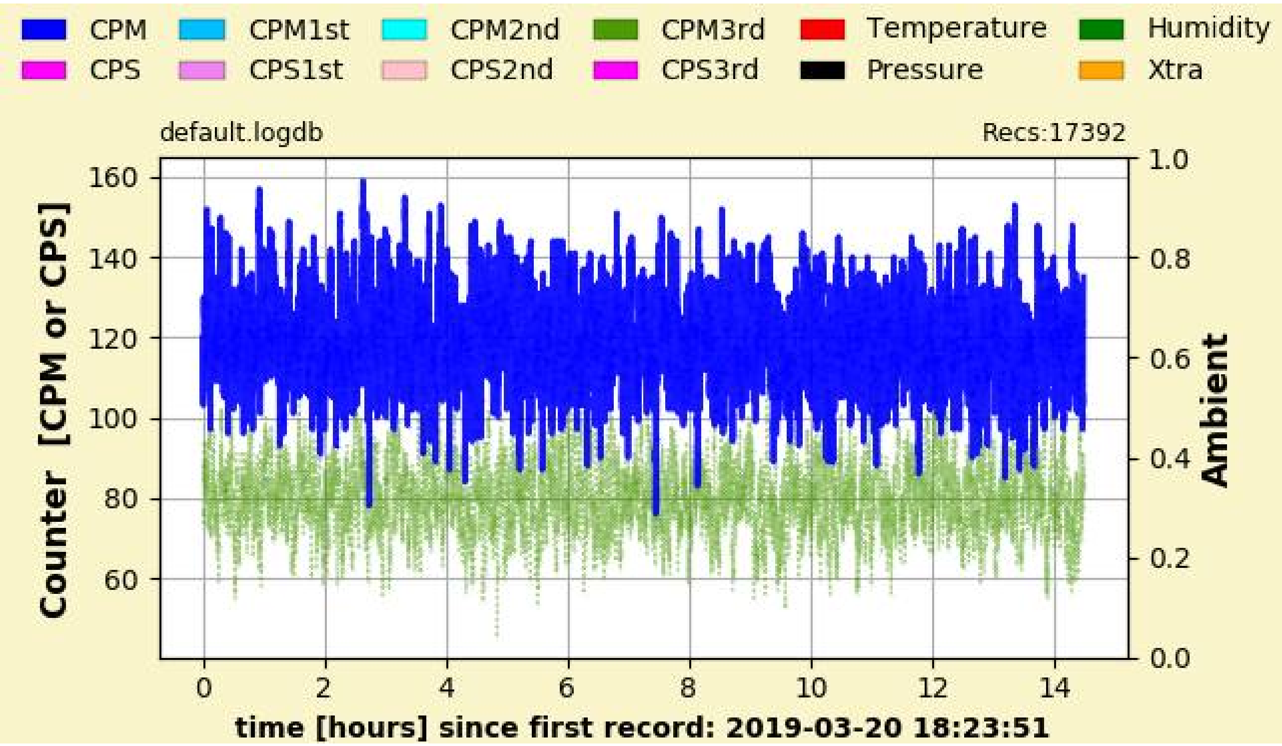


Figure 29: GMC-300E+ (blue) digital recording, and SGP-001 Device (green) audio recording

27 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

28 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

29 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

RadMon Devices

The RadMon devices [30]) 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 [31]) 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 [32]) for temperature, barometric-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 [33]) – 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, barometric-pressure, and humidity from its BOSCH BME280 sensor. All 4 values are then sent through my wireless home network to my

30 DIYGeigerCounter https://sites.google.com/site/diygeigercounter/

31 https://en.wikipedia.org/wiki/Internet_of_things

32 BOSCH BME280

33 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.

router and then to a broker server located in North America [34]). The total of reading the 3 environmental 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 [35]). 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 definitively want to have the manual [36]) 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 30. For the rest you can follow the instructions on the smartphone. When done, press the RST button on the RadMon+.

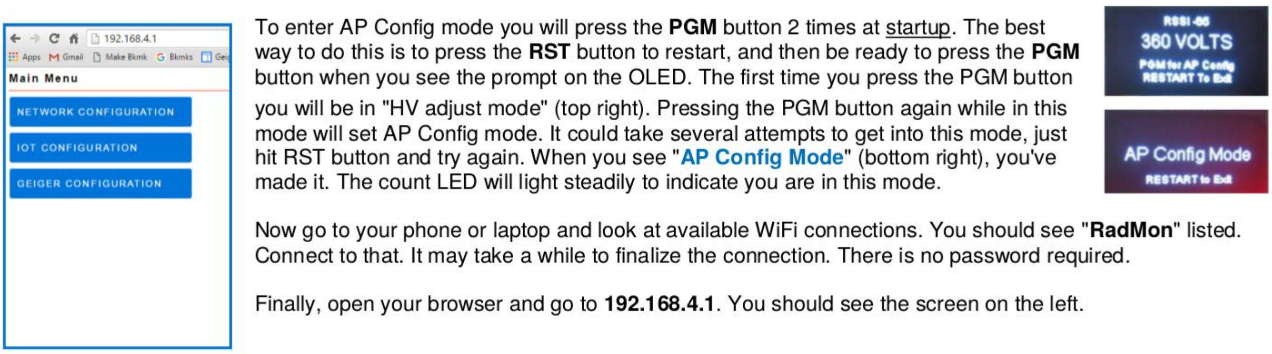


Figure 30: Activating RadMon Configuration on a Smartphone

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

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

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

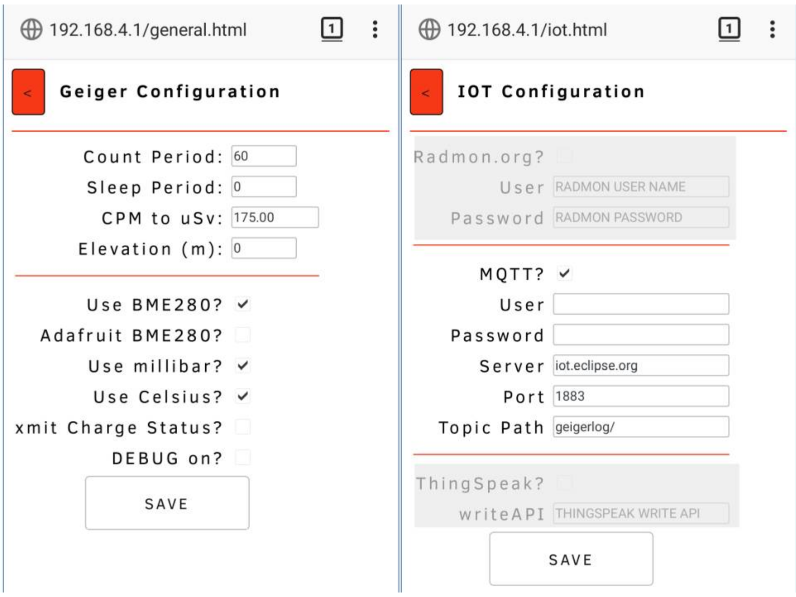


Figure 31: RadMon+ Configuration pages

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 31 as screen shots from my smartphone:

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

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

Figure 32 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.

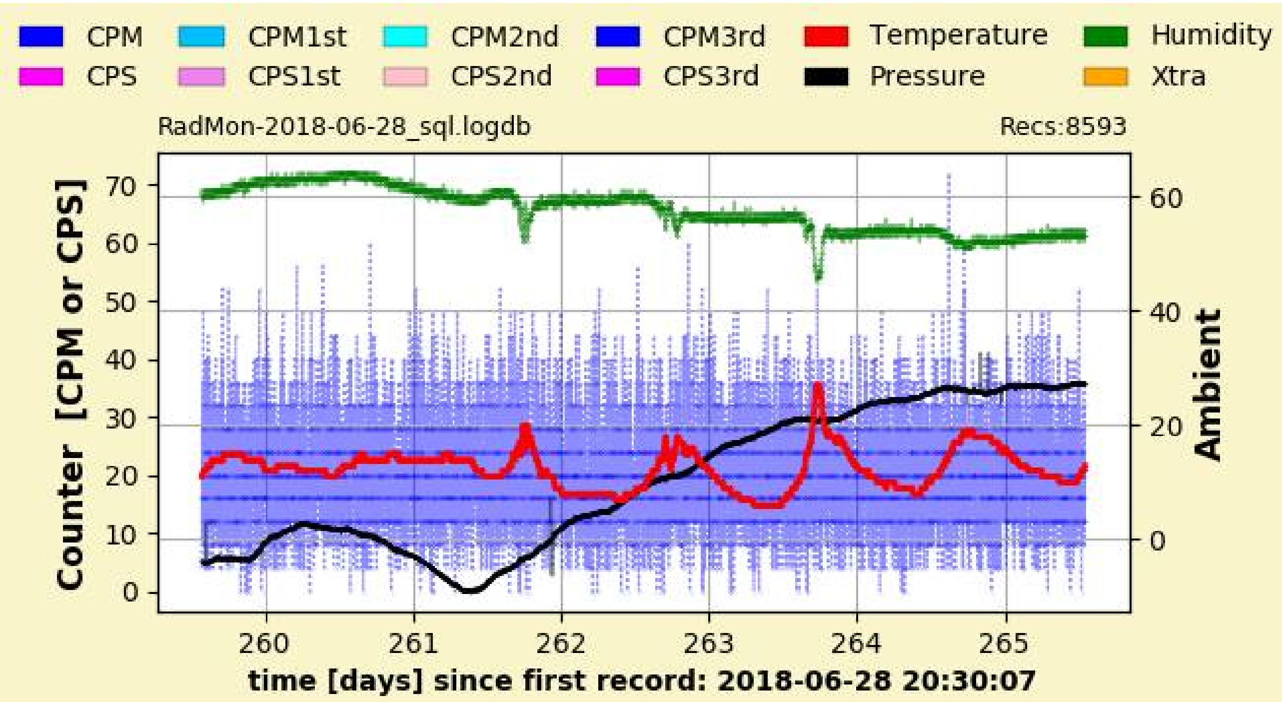


Figure 32: Long-Term Recording from an Outside RadMon+ Weather Station

Gamma-Scout Devices

The company Gamma-Scout (https://www.gamma-scout.com/en/) offers these four Geiger counter models: ***Standard***, ***Alert***, ***Rechargeable,*** *and* ***Online*** [37]). From left-to-right the amazon price goes up from about 380€ to 525€ (Feb 2021).

Some hardware and firmware choices are unexpected, so will be explained below in more detail. In brief:

- All have the same type of tube, which can detect alpha, beta, and gamma radiation
- All counters allow to store a history on board, which can be downloaded to a computer
- Logging is possible **only** with the 'Online' model, and only with a cable connection
- There is no WiFi on any of these models

GeigerLog supports all models and all options, with abilities beyond what the included manufacturer's software can do!

Hardware and Firmware

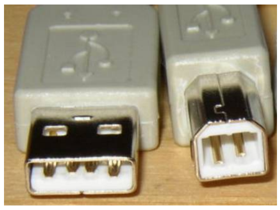
The tube

The tube of all Gamma-Scouts is an LND 712 [38]), a small tube roughly the size of a kid's thumb (49.2x15.1mm), with a sensitivity of 108 CPM/(µSv/h) for Co60, as rated by the manufacturer.

The advantage of this tube is that it has a Mica window, so it can detect alpha-particles, but the window is small with a diameter of only 9.1mm!

The counters have a convenient lever to move different shields in front of the tube's window which leaves it either completely open, blocks only alphas, or also blocks betas.

The USB-Port



The connection between computer and counter is made with an USB cable with a Type A plug on one side, and a Type B plug (the squarely one) on the other side. But electrically the connection is only a serial connection, similar to what the GMC and I2C devices use. The chip doing the conversion between USB and Serial signals is an FTDI chip (Vendor=0x0403, Product=0xd678). This chip supports USB2.0, but only in its slow mode of 12 MBit/s, which is not faster than USB1.0. But as the serial connection speed even at 460 800 baud (=0.46 MBit/s) is only a fraction of that, the chip is fast enough.

And while the serial speed is fast for a serial connection, only the Online model is that fast, the other 3 models offer only rather ancient 9600 baud!

37 https://www.gamma-scout.com/en/measures-radioactivity-easily-and-reliably/

38 https://www.lndinc.com/products/geiger-mueller-tubes/712/

To make matters worse, the firmware uses a 7-bit transfer mode, i.e. a single byte of 8 bit is split into two ASCII-coded bytes of 7 bit. Thus the effective transfer rate is further reduced two-fold to now 4800 baud and 230 400 baud, resp.!

Logging

The "Gamma-Scout Online" is the latest model and the only one of the four to offer an option for logging. Basically, the counter is set into a mode where it sends out count information at regular intervals. This is the only existing mode for logging.

The interval can be set to 2, 10, 30, 60, 120, and 300 seconds, so the returned data mean Counts-Per-2seconds, Counts-Per-10seconds, …, Counts-Per-300seconds. GeigerLog can handle the settings and collect, record, and display the data. As long as the interval is 60 sec or less, GeigerLog also adds the CPM data by summing up the last 60 seconds worth of counts. This CPM is true Poissonian! For any longer interval there is obviously no Poissonian way back to CPM, but you can always scale the data (e.g. divide the 300sec data by 5) by using GeigerLog's Scaling option (see ValueScaling and GraphScaling on page 30). Though then this no longer forms a Poisson distribution.

This Gamma-Scout "Online" mode is the same as the "heartbeat" mode found in GMC counters, which, however, is outdated and not used by GeigerLog, as the GMC's send-on-demand mode is easier to handle and much more useful.

History Memory

All 4 devices have an internal history memory of only 64k Bytes, which makes the slow transfer speed a bit more bearable ;-). All counts are stored as a 2 bytes floating point value as described in the 'Gamma-Scout Communication Interface' article [39]). It uses an 11-bit mantissa, and a 5-bit exponent.

The disadvantage is that 11 bits can hold values only up to 2048, so everything greater will loose precision, and while everything smaller than 255 could be coded with just a single byte, it now needs the additional, empty second byte. The advantage is that you always know how many bytes are needed for storage independent of count rate, and do not have the overhead of distinguishing between 1,2,3, and perhaps 4 byte values! This latter problem is a significant one with the algorithm used in the GMC counters (see Appendix E – GMC Device: Internal Memory, Storage Format and Parsing Strategy on page 120).

The counts will be collected over a selectable interval ranging from 10 seconds to 1 week. Note that anything faster is not possible, in particular CPS (Counts Per 1second) can NOT be set! Strangely, the 2 second interval, while available for logging, is not available for the history interval setting!

So the memory can hold some r00 values, plus some house keeping information like time stamps, intervals, alarm reports, etc. Thus with a 10 sec interval the memory is filled after about 3 days. With a 1 week interval the memory is filled after 615 years! I wonder how enticing this is for people to purchase this counter? And I wonder what they say when they have to replace the battery "after a few years" and all data in the memory are lost?

39 https://www.gamma-scout.com/wp-content/uploads/Gamma-Scout_Communication_Interface_V1.7.txt

Operating the Gamma-Scout Counter

Installation

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

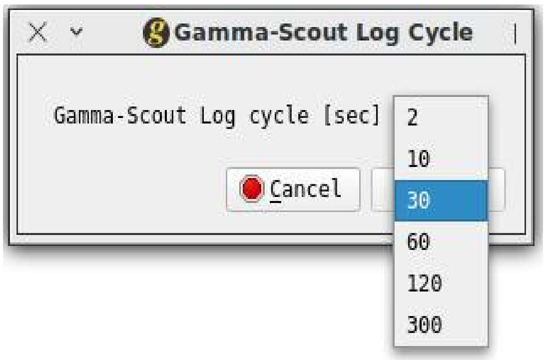
See chapter Appendix B – Connecting Device and Computer using a Serial Connection on page 109 for a detailed explanation!

Since the Gamma-Scout device identifies itself with its name (like simply: GammaScout USB) it is easy to select the proper port with a baudrate specific for this device from menu **Tools** → **Set Port …**, or by clicking the **Set Port** toolbar icon.

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 these settings and insert them into section `GammaScoutSerialPort` of your GeigerLog configuration file `geigerlog.cfg`.

Logging

The Gamma-Scout Online must be explicitly set into its Online mode for logging. Otherwise only empty values will be collected. From the menu select: **Device** → **Gamma-Scout Series** → **Set to Online Mode**. This dialogue pops up and allows you to select the interval for logging.



The GeigerLog log cycle should be set to the same setting as the Gamma Scout. It can be set faster, but not slower. Then start logging.

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 as Gamma-Scout calls a History Download) 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 it were downloaded from a Gamma-Scout device.

Data Interpretation

The Gamma-Scout devices do produce only “Counts-Per-Interval” data, with 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 interval is set at 1 minute!

By default GeigerLog maps the CPInterval counts to variable CPS3rd. If the interval is 60 sec or less, then GeigerLog creates a true CPM column mapped to variable CPM3rd by summing up the latest 60 seconds worth of counts. The interval itself is mapped to variable X.

Next lines are an excerpt of logging data done with a Gamma-Scout interval of 2 seconds. So column CPS3rd contains “Counts-Per-2seconds”, while CPM3rd contains true CPM. Of course, it takes 30 records of 2 sec each before a full 1 min of data can be summed up for the 1st CPM value!

```
#>  Index,            DateTime,  CPM3rd,  CPS3rd,       X
#  HEADER, 2021-02-20 13:15:47, LogFile newly created as 'default.logdb'
# DEVICES, 2021-02-20 13:15:47, Connected: Gamma-Scout : CPM3rd CPS3rd X
# LOGGING, 2021-02-20 13:15:47, Start: Cycle: 2 sec
        1, 2021-02-20 13:15:49,        ,     0.0,     2.0
        2, 2021-02-20 13:15:51,        ,     0.0,     2.0
        3, 2021-02-20 13:15:53,        ,     0.0,     2.0
        ...
       28, 2021-02-20 13:16:43,        ,     0.0,     2.0
       29, 2021-02-20 13:16:45,        ,     0.0,     2.0
       30, 2021-02-20 13:16:47,    14.0,     1.0,     2.0
       31, 2021-02-20 13:16:49,    14.0,     0.0,     2.0
#>  Index,            DateTime,  CPM3rd,  CPS3rd,       X
```

Both columns CPM3rd and CPS3rd can be Poisson tested, and should give a proper distribution. The above data as a graph are shown in Figure 33.

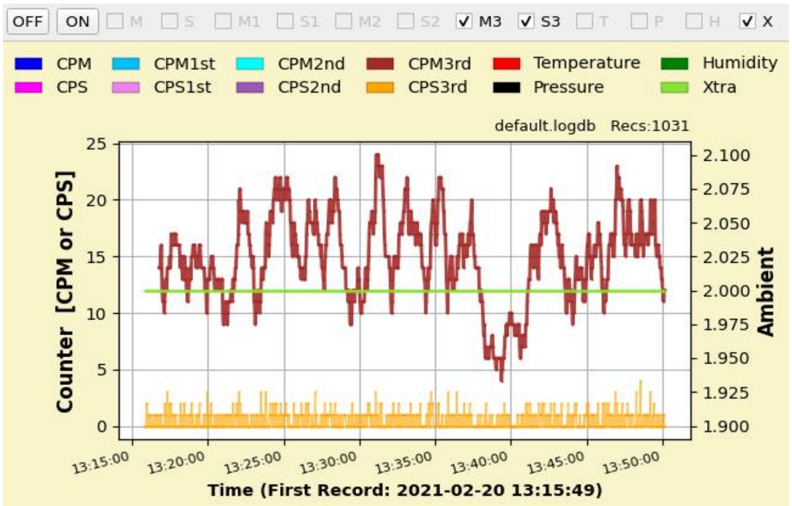


Figure 33: Gamma-Scout Online model used for logging with a 2 sec interval

Orange: the “Counts-Per-2sec” data as delivered by the counter, brown: the CPM data calculated by GeigerLog, green: the interval of 2 sec (Ambient scale)

Counts vs. µSv/h

The conversion between a dose rate in Counts and in µSv/h is based on either CPM or CPS. Since the Gamma-Scout devices do not offer a 1 second interval, all CPS data shown as µSv/h will be too high by a factor equal to interval! CPM data calculated by GeigerLog will convert properly.

A 3-day history download from a Gamma-Scout Online device, recorded with a 10 sec interval is shown in Figure 34.

Making the Poisson tests on both data sets from Figure 34 gives the results shown in Figure 35. Both data sets are fully Poissonian.

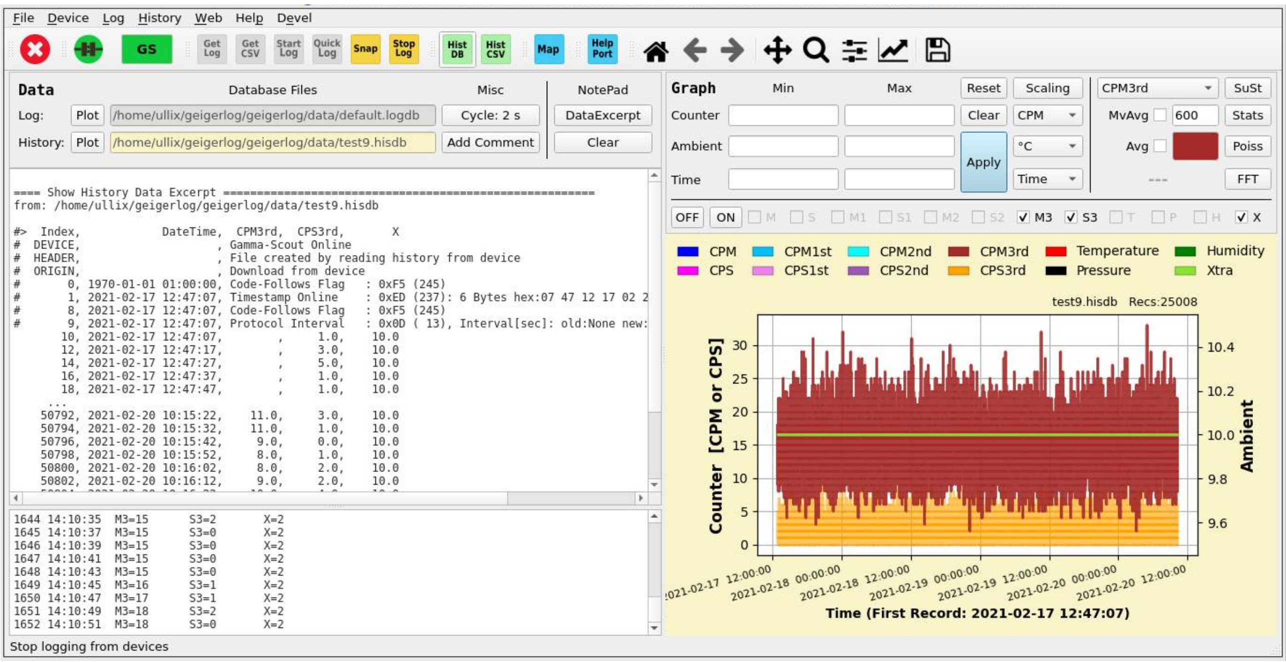


Figure 34: A 3-day history download from a Gamma-Scout Online device

The recording interval was 10 sec. CP10sec: orange, CPM:brown, X (Interval):green

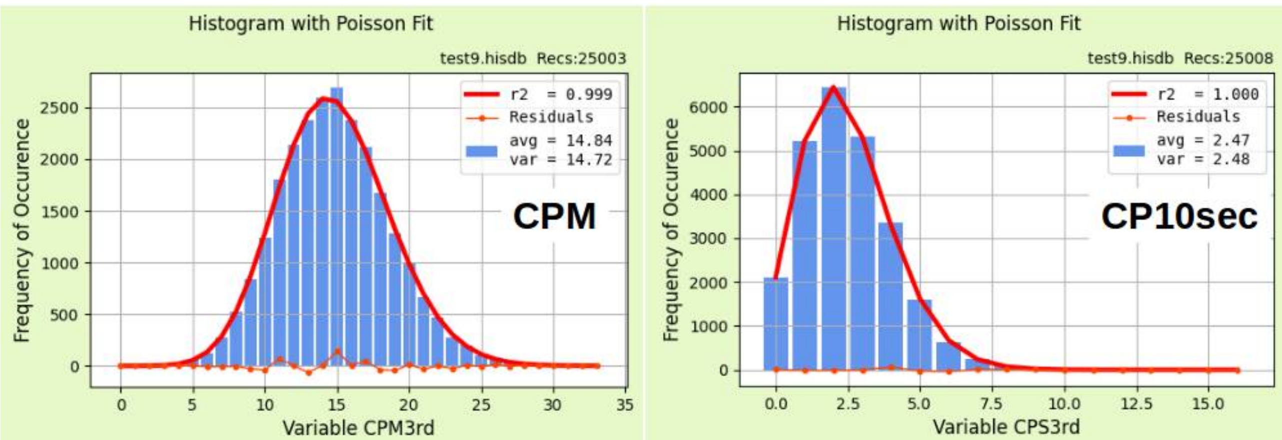


Figure 35: Poisson Tests on the data shown in above figure

I2C Devices

GeigerLog can handle I2C based devices.

Presently it supports these I2C sensors: **LM75** (temperature), **BME280** (temperature, barometric-pressure, humidity), **TSL2591** (light sensor visible, infrared), **SCD30** (CO2, temperature, humidity), **SCD41** (CO2, temperature, humidity).

Such sensors require an I2C connection, which is not available on today's computers, at least not for regular users. However, hardware is available, so called **USB dongles**, which provide this type of connection.

I have selected 4 different dongles and evaluated their performance with each of the I2C devices listed above within GeigerLog. The result is published in article **"Review of USB-To-I2C Dongles as used by GeigerLog"** available in my article folder [40]). A follow-up assessment of devices with respect to measurement of CO2 is published in article **"Measuring CO2 with SCD30, SCD41, and MiniMon"** in the same article folder.

While GeigerLog can support all dongles and devices, the clear dongle champion **USB-ISS** is chosen for future work. Its advantages:

- worked with all I2C devices tested
- the fastest of all the dongles by a good margin (up to 20 fold faster!)
- cross-platform compatible (Linux, Windows, Mac, ...)
- no driver installation by user (installed by default; except perhaps in old Windows systems)
- auto-configurable in GeigerLog
- similar or lower price than the competitors
- other technical advantages (supports 3.3V, bus speed up to 1MHz, easy to program)

Note: The 2nd fasted dongle is the ELV dongle, also cross-platform usable without user installations. The 3rd fastest is the IOW dongle, so far working only on Linux, and requiring a driver installation [41]). The 4th one is the FTD; it was removed from GeigerLog, because it failed to support one I2C device, it was difficult to install, and it was awfully slow.

Connecting

GeigerLog, beginning with its version 1.3.0, can auto-detect the presence of an USB-to-I2C dongle USB-ISS and make the correct settings, thus providing for **plug-and-play use of USB-ISS** dongles! [42]). GeigerLog will auto-configure it for best performance.

If this fails then you probably have a computer with an older Windows without the pre-installed driver. Look for installation files for a "Microchip Technology, Inc., Devantech USB-ISS" chip (idVendor=0x04d8, idProduct=0xffee) and install.

40 https://sourceforge.net/projects/geigerlog/files/Articles/

41 Follow the install instructions for libiowkit-1.X https://www.codemercs.com/downloads/iowarrior/IO-Warrior_SDK_linux.zip

42 This plug-and-play will also work for the other dongles, however, in the case of IOW and FTD you'll first have to install the required, non-standard driver!

Then you need to wire-connect your I2C devices to the dongle, configure them in the GeigerLog configuration file `geigerlog.cfg`, start GeigerLog and select menu **Device→Connect Devices** as explained in Establishing a GeigerLog Connection to Your Devices on page 16.

Example Setup

This example is taken from article **"Measuring CO2 with SCD30, SCD41, and MiniMon"** [43]**.** The I2C devices **SCD30**, **SCD41**, **BME280** were mounted on a breadboard, and connected to GeigerLog using the **USB-ISS** dongle. The assembly is seen in Figure 36.

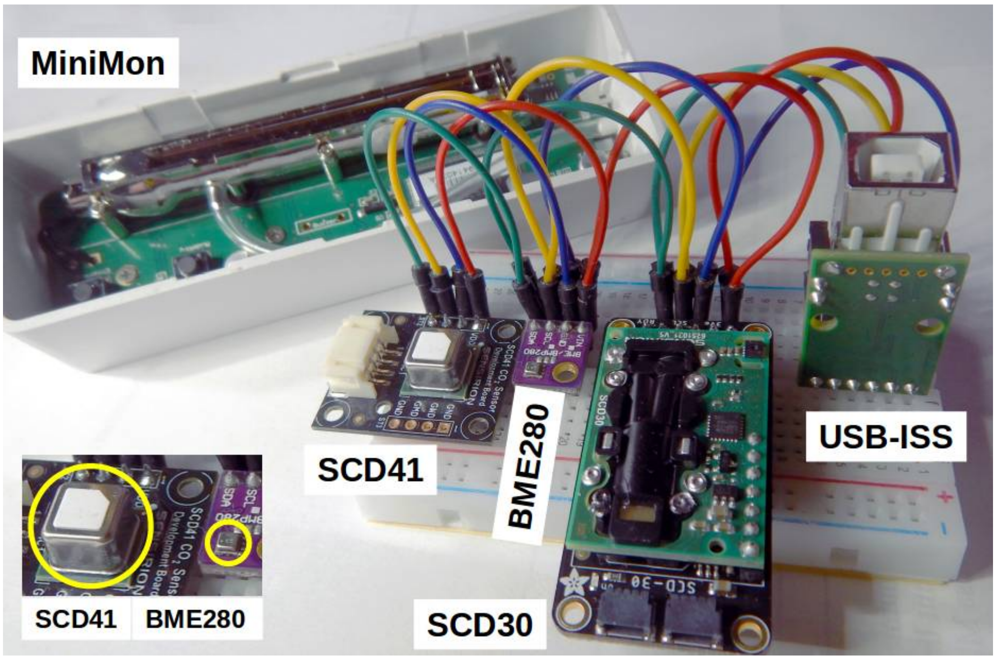


Figure 36: The assembly of the devices

From left-to-right you see the MiniMon (with its backside removed), the SCD41, the BME280, the SCD30, and the USB-ISS dongle, with its USB-Type-B receptacle pointing upwards. The inset on the bottom-left serves as a size comparison for the SCD41 and BME280. The SCD41 is small, but the BME280 is tiny!

Results from a 30 h run of this assembly with GeigerLog is shown in Figure 37. The bottom 3 curves are CO2 data in ppm, the top curves are Temperature data (values scaled by 50x) in °C. A detailed discussion is in the referred to articles [40)].

These I2C devices are fast, each responding within 2 … 6 milliseconds when on the USB-ISS!

43 Download from SourceForge folder: https://sourceforge.net/projects/geigerlog/files/Articles/

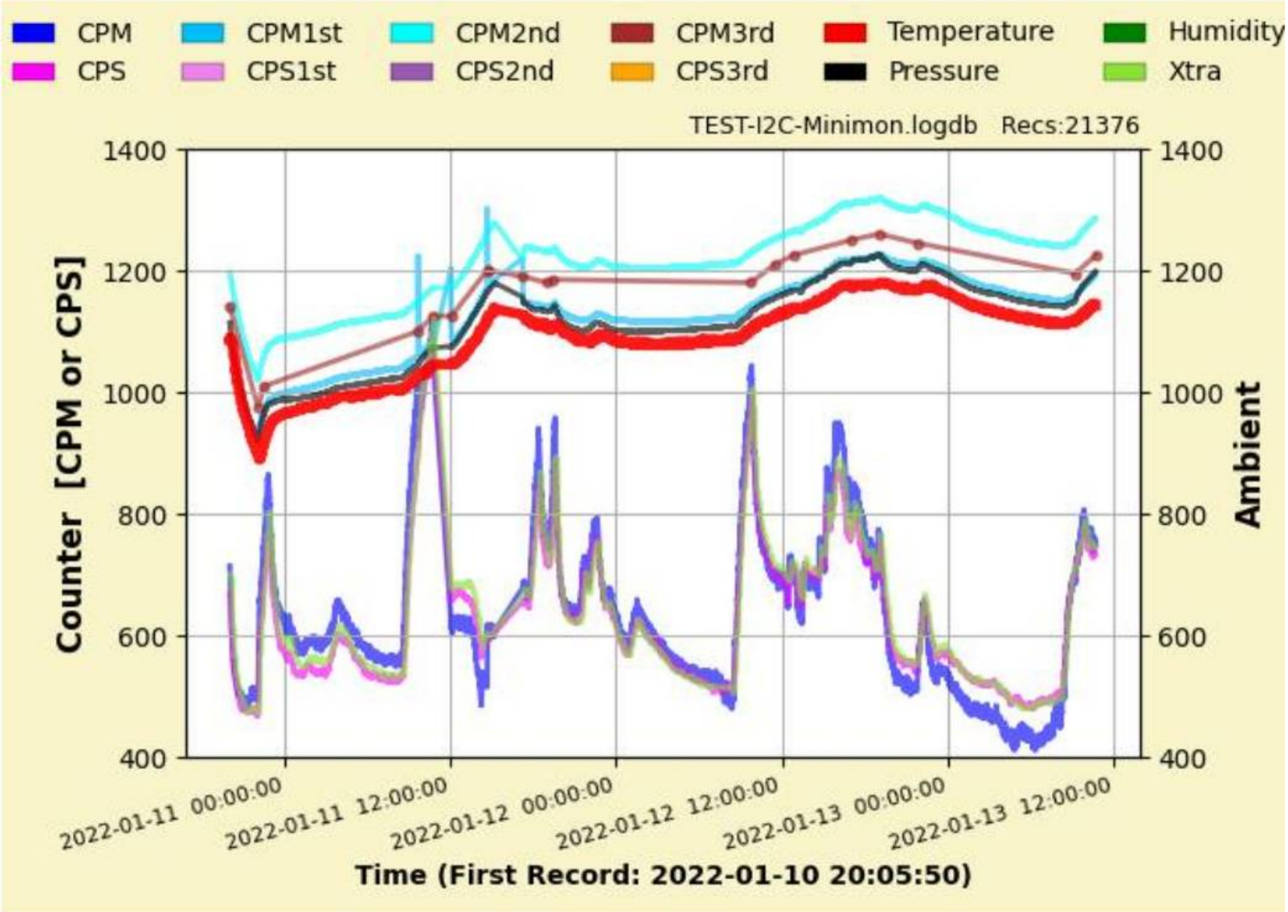


Figure 37: A run with I2C devices BME280, SCD30, SCD40 via USB-ISS, MiniMon and Manu

LabJack Devices

Running this device with GeigerLog is currently supported on Linux and it may work on Mac, but will NOT work on Windows!

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.

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

Installation

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 2.0.4 is available using pip, see Appendix H – Installation on page 135.

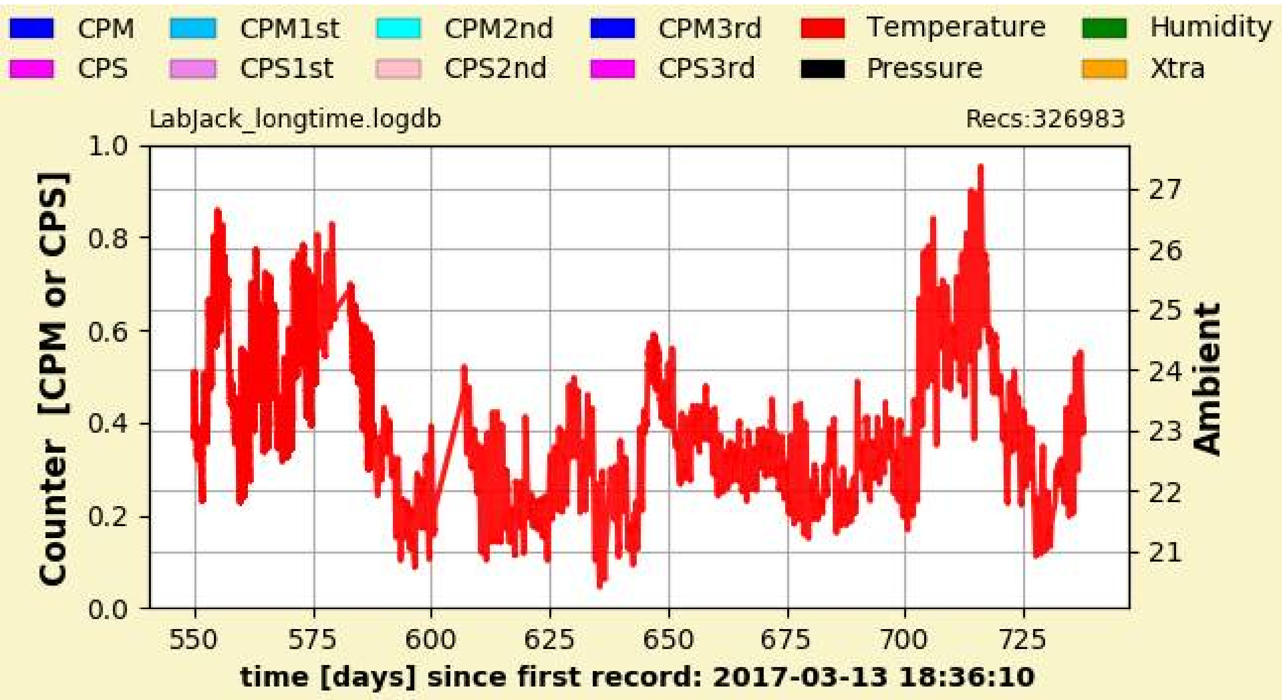


Figure 38: Long-Term Recording – 2 years – with the LabJack, showing Temperature only

MiniMon Devices - CO_2 Monitor

This device can be used with GeigerLog only when run on a Linux operating system!

This device and look-alikes are available from several companies. Mine is from **TFA-Dostmann** [44]). The original manufacturer may be **ZyAura** [45]); a (limited) datasheet is here [46]).

There is no brand-name for this device, so I created the name **MiniMon**.

Technology

MiniMon measures CO2 based on the Non-Dispersive-Infrared (NDIR) technology, which is a well established standard for measuring CO2. It uses an LED generated infrared beam, and measures the absorption of certain CO2-specific wavelengths in the air sample. The result is presented as ppm (parts per million).

For reference:

- outside air: CO2 in the low 400 ppm
- class-rooms, offices: CO2 1000-2000 ppm
- air exhaled by mammals: CO2 40 000 ppm (4%)
- Rooms, with people working in them, should be vented when CO2 reaches 1000 ppm
- The US Navy permits up to 7000 ppm on its submarines for combat-ready sailors.

MiniMon displays the CO2 concentration on a small B&W LCD screen, alternating with the display of temperature in °C. Some devices (not mine) also display relative humidity.

Installation

There is nothing to install. MiniMon has no buttons or switches and only a single plug for a micro-USB cable, which is used for power supply and data delivery. It has 3 LEDs in green, yellow, and red which are lit to indicate air quality. It is easily made a mobile device by connecting with a power bank – I used it with a 13000 mAh power bank, which was still 70% full after 1 week!

Connecting

MiniMon is used with the Linux HIDRAW (**H**uman-**I**nterface-**D**evice **Raw**) driver, which is the kernel interface for Raw Access to Human Interface Devices. These drivers are **specific to Linux**.

Once you have found MiniMon's hidraw address – explained next – you enter it into the GeigerLog configuration file in section `MiniMon` as parameter to `MiniMonOS_Device`. It will be something like `/dev/hidraw5`.

44 https://www.tfa-dostmann.de/

45 https://www.zyaura.com/product-detail/zgm053u/

46 https://www.zyaura.com/support-download/manual-zgm053u/

However, an easier way is to create a udev rule, so that you find MiniMon always under the same address, e.g. `/dev/minimon`, and can leave parameter `MiniMonOS_Device` on `auto`.

MiniMon Address – per MANUAL setting

Without plugging-in MiniMon issue the command:

```
ls -al /dev/hidraw*
```

Now plug-in MiniMon, and issue the same command. The newly appearing line, showing e.g. `/dev/hidraw5`, provides the needed hidraw address for MiniMon.

Depending on permission settings on your computer (explained in Appendix C – HOWTO deal with read and write permissions for the serial port when on Linux), you may also have to give this command:

```
sudo chmod 666 /dev/hidraw5
```

MiniMon Address – per UDEV rule (recommended)

With root permission add file `90-co2mini.rules` (included in the GeigerLog package) into folder `/etc/udev/rules.d`. To activate this udev rule, issue command:

```
sudo udevadm control –reload-rules
```

Then unplug and replug MiniMon. Your device will now always have address `/dev/minimon`.

NOTE: If you have multiple MiniMons and want to run them all simultaneously, read the file `90-co2mini.rules` for the changes to make to this udev rule.

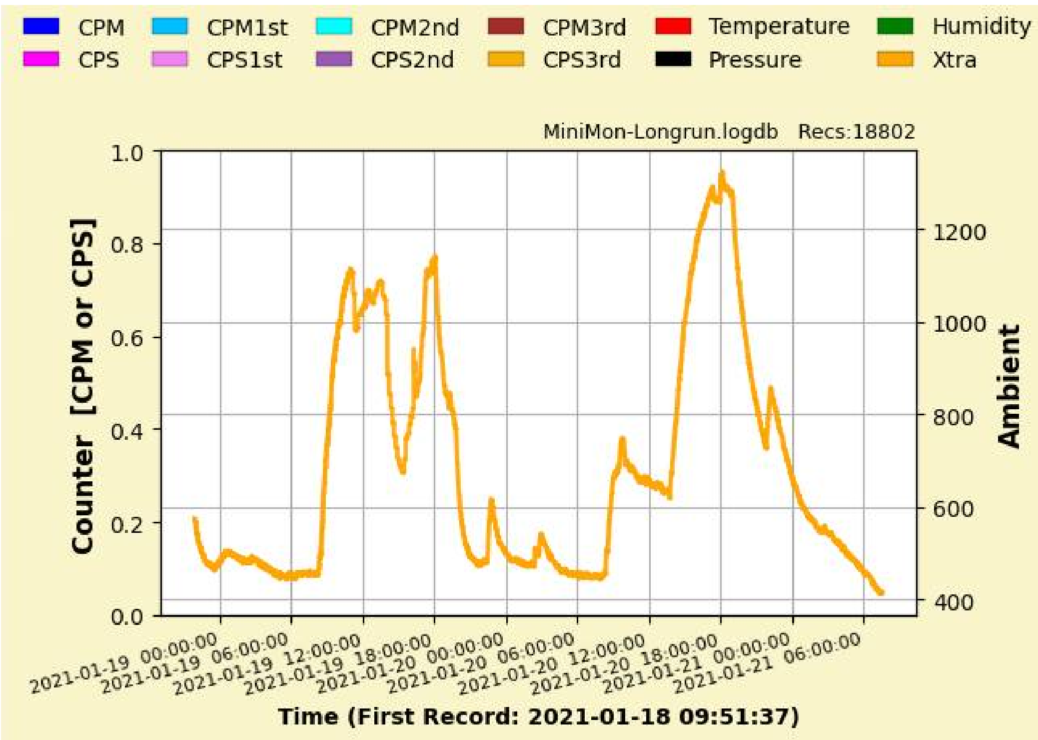


Figure 39: Measuring CO2 Concentration in an office

Logging Data

By default GeigerLog will read Temp (for temperature) and Xtra (for CO2) from MiniMon. You could also activate Humid (for humidity), but unless your device does have a humidity sensor, you would only read 0 (zero).

Figure 39 shows just the CO2 recording of an example run taken in my office. There is high-CO2 (~1000 ppm) during my working hours, fluctuating depending on opened or closed doors/windows, and going down to near 400 ppm during the night.

An in-depth comparison of this MiniMon device with two I2C based devices (SCD30, SCD41) is found in my article "**Measuring CO2 with SCD30, SCD41, and MiniMon**" [47]).

For more on CO2 measurement see also in this manual: I2C Devices on page 64.

History Download

The device has no internal memory, so a history download is not possible.

47 https://sourceforge.net/projects/geigerlog/files/Articles/Measuring%20CO2%20with%20SCD30%2C%20SCD41%2C%20and%20MiniMon-v1.0.pdf/download

Simul Device

This is a device which simulates a Geiger counter. The data are generated as CPS through a **mathematically defined Poisson distribution**. GeigerLog calculates the CPM values by summing up the last 60 sec of these CPS data.

The purpose of this pseudo-device is to run some tests knowing that you have precisely defined records. Sometimes it is helpful just to remind yourself of how crooked a time course of counts can look like, and still represent a perfectly good Poisson distribution!

Installation

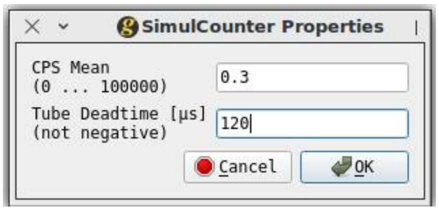
There is nothing to install. Simply activate the Simul Device in the `Simul` Device section of the GeigerLog configuration file `geigerlog.cfg`.

Settings

The data will be mapped to CPM and / or CPS. No other variables can be used.

While the other variables can actually be defined for the Simul Device, those will have different meanings. This is explained elsewhere and can be made available if anyone wants to experiment with the Simul Device.

Other settings are:



- Count-rate Mean; default is CPS=0.3
- Tube Sensitivity; default is 154 CPM/ (µSv/h)
  The simulated counts can be treated like true “counts”
- Tube Deadtime: default is 120 µs
  Becomes relevant only in advanced experiments which
  use high count rates and need to account for deadtime losses.

Mean and deadtime can be changed during a run by calling from the menu **Device → Simul Device Series → Set Properties**.

An article on deadtime correction will be published soon on the GeigerLog Sourceforge side, which will apply those other functions of the Simul Device.

Simul Device Test-run

A run with the Simul Device is shown in Figure 40. Of course, CPM needs 60 sec before it can report its level. And despite the Poisson curve looking a bit crooked (of course using only the part after 1 min!), it its a perfect Poisson distribution. It is statistics, after all, which makes it all smooth eventually!

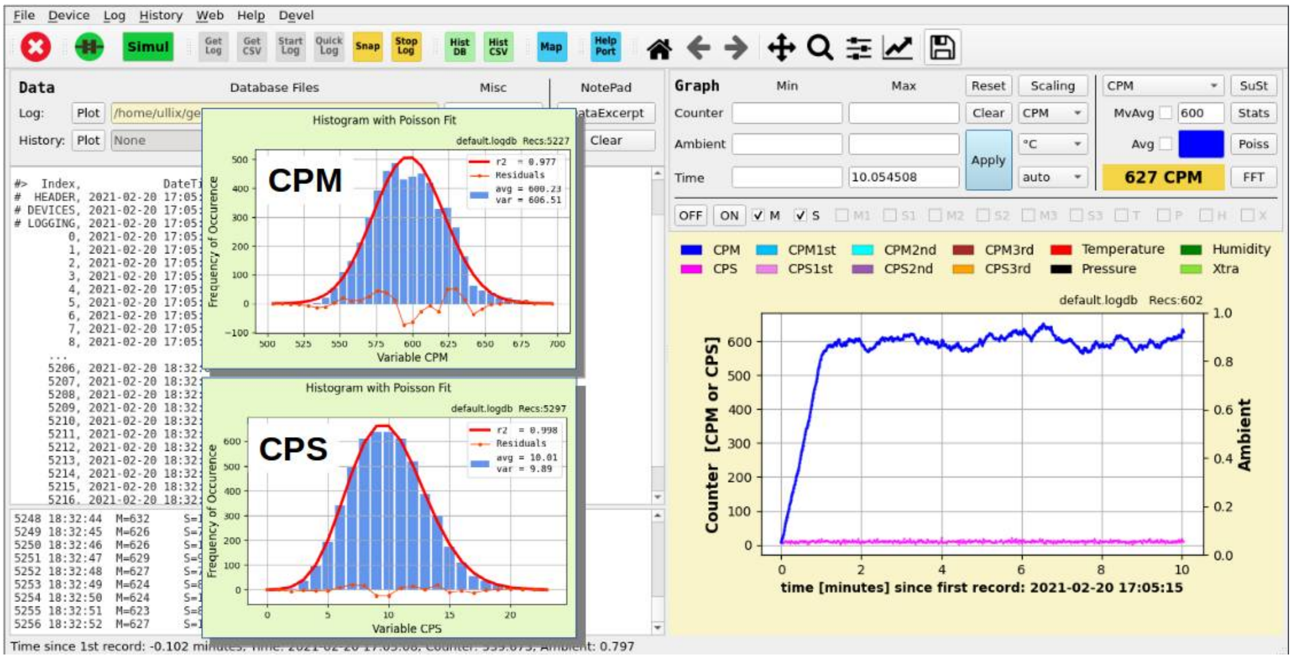


Figure 40: A run with the Simul Device

Manu Devices

Manu is a pseudo-device which allows to enter numeric data manually, which then will be shown in the graph just like any other data. This is helpful to bring data into GeigerLog which you can't (easily) enter electronically, but want to compare them directly with other data in GeigerLog.

Examples could be:

- Study the influence of the distance between Geiger counter and radioactive source. Enter the distance via the Manu device e.g. as variable Xtra.
- Measure the Geiger tube's response to temperature. You may have a remote Temp-Sensor but it has only a manual readout (like a meat-temperature sensor for the barbecue). Enter the temp via the Manu device e.g. as variable Temp.
- You measure the capacity of a LiIon battery powering a microchip. The chip uses its own ADC to report a voltage via the GeigerLog **WiFiServer device**, and you use a precision DVM to measure true voltage manually and enter it to the **Manu device**.

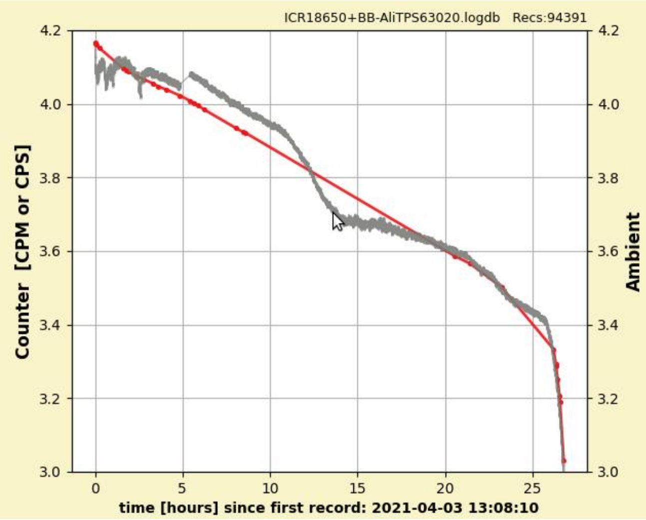


*Figure 41: Measuring battery voltage using GeigerLog **WiFiServer** and **Manu***

See text

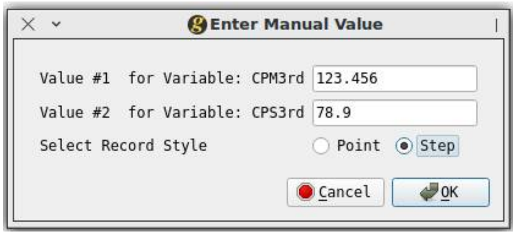
Figure 41 shows the result of the latter experiment. In gray the online-ADC voltage, in red the DVM voltage, using the variables CPM3rd and Press (my motivation for the Manu device).

Installation

There is nothing to install. Simply activate the **Manu** device in the **[Manu]** section of the GeigerLog configuration file `geigerlog.cfg` and define the variables you want to use.

Settings

Click menu **Device → Manu Series – Enter Values Manually.** In the dialogue you can enter values for all the variables you have configured for Manu, which can be up to the full set of 12 variables. The dialogue will expand to provide entry fields for all variables.



Choosing **Step** will keep the value entered for all next log cycles, appropriate when the value remains constant, like for the source-distance experiment described above. Using **Point** creates single entries, which GeigerLog connects with a line, like the red curve of Figure 41.

WiFi Devices managed by GeigerLog

This applies to all web enabled devices. They could be connected either by wire (LAN cable) or wirelessly (WiFi). However, it seems highly unlikely that devices as discussed here will ever be connected by a LAN cable; so we'll limit the discussion to **WiFi connections only**.

Distance

The WiFi devices will be connected to a local WiFi, and those as well as GeigerLog can freely roam within the reach of this WiFi net. To support a greater distance between device and GeigerLog, WiFi range extenders can be used like with any other WiFi enabled device. For even greater distance your router can be set up with port forwarding technology, so that calls can even go via the internet to any place on the planet.

Speed

The devices show up in my FritzBox Mesh with speeds between 60 … 80 Mbit/s. This is some 500 to 800 times fast than current connections over USB-To-Serial! A download of the 1 Mbyte memory of some counters would not take more than 3 min, but under 3 sec!

Power

One disadvantage is that WiFi needs more power than a USB / USB-to-Serial connection. In addition, the cable-connected device was charged while it was connected. Now it may have to rely more or even solely on its own battery.

However, other power delivery methods can be used, like USB Wall Chargers, solar power, power bars and others.

WiFi Devices Types

There are two types of WiFi devices possible:

- **WiFiClient Devices** (Example: GQ's GMC counters (when WiFi enabled)
- **WiFiServer Devices** (Example: AmbioMon Devices)

They will be explained next.

WiFiClient Devices

These are external devices which act as **WiFiClients** and call the web server built into GeigerLog.

Example: GMC counters act as **WiFiClients** when sending data via WiFi!

WiFiClients operate on their own schedule, as set by the user. GeigerLog, like any other server, is waiting patiently for such a call, and becomes active only when such a call arrives. The GMC counters, as example, have a default setting of once every 2 min for calling the gmcmap.com server and updating the data.

To transfer data to the server, the WiFiClient has to wrap the data values into the URL used to call the server. In web-lingo it is creating a GET call [48]).

The situation is basically the same as if you were typing a search request into the search field of your browser and then send it to the desired search engine. The search engine will look at the things you typed in your request and returns a web page as response, and then waits for the next call.

When GeigerLog receives a call, it extracts the data and answers with only a simple response, like "OK". This is the complete response "web-page". In essence, the WiFiClient has uploaded data to GeigerLog by seemingly requesting a web page, in which the WiFiClient is not interested in (and GeigerLog not delivering).

Otherwise the data are being handled by GeigerLog just like any other data, they will be saved, printed, displayed, scaled, and plotted.

Even the GMC counters calls can be redirected into GeigerLog, but it does take some extra effort. It will be shown below.

Installation

In the `[WiFiClientDevice]` section of the GeigerLog configuration file `geigerlog.cfg` activate the **WiFiClient** device, and change other settings as appropriate:

Any **WiFiClient Device** needs to know the IP address associated with the computer running GeigerLog. If you don't know already, GeigerLog knows its IP Address and tells you when you click menu **Web→Show IP Status** or the **IP** icon in the toolbar**.** This is set by your home network and cannot be changed from within GeigerLog.

The device also needs to know the port number on which GeigerLog is listening. The default is **8000**; change if needed.

Further, look at tube sensitivity and variables and change as needed.

Device Type

GeigerLog supports two types of WiFiClient Devices. Only one can be active in a run; any calls from the other one will be rejected by GeigerLog.

48 https://www.w3schools.com/tags/ref_httpmethods.asp

Type 1: GENERIC:

As the name implies, this is the more universal type. The external device assembles all data into a GET command. Assuming a device which wants to transmit CPM=99, CPS=3, and Temp=24.6, this URL takes the form (for the M, S, … codes see footnote [52]):
http://<GeigerLogIP>:<GeigerLogPort>/GENERIC/?M=99&S=3&T=24.6

or, using all 12 variables with simple values and GeigerLog IP=10.0.0.20, GeigerLog Port=8000:
http://10.0.0.20:8000/GENERIC/?&M=1&S=2&M1=3&S1=4&M2=5&S2=6&M3=7&S3=8&T=9&P=A&H=B&X=C

GeigerLog will answer with:
OK for Generic WiFiClient Device

A Python file **`DemoWiFiClient.py`** is available in the GeigerLog folder **`gweb`**, which can be run to simulate a Generic WiFiClient Device.

Type 2: GMC:

This mode accommodates the **WiFi-enabled GMC counters** by imposing certain restrictions to the GENERIC Type. The default behavior of the GMC counters is explained in chapter WiFi Equipped GMC Counter on page 54.

In GeigerLog the AID and GID parameters are ignored, can have any value, and may even be absent. Of the 3 variables: CPM, ACPM, uSV [49]) only CPM matters, because the others can be calculated by GeigerLog. Nevertheless, all 3 variables can be recorded in GeigerLog. The counter limits data update to no more than once per minute, so CPM is the only meaningful variable to send.

However, these counters have a bug and a deficiency in the counter's firmware, and their use with GeigerLog requires a **URL redirection**, which is explained next.

Redirection for GMC counters

Deficiency – Port Number Problem: The firmware of a GMC counter forces the use of port=80. However, port numbers 0 … 1024 are reserved for users with Admin rights, and for security reasons it would be a bad idea to run GeigerLog as Admin, solely to open such a server port! Perhaps even in conflict with an active, regular web server on your PC, which almost always also listens at port=80.

Bug – "Unsafe" Operation Problem: The GMC counter's firmware form the URL in such a way that modern servers reject their contact as a security issue. One can overcome this by explicitly putting an "Unsafe" into their configuration file [50]). But which Web server Admin would convert their server intentionally into a security risk?

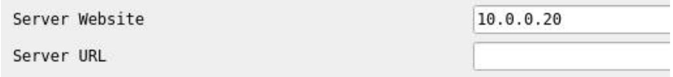
Since I have a full **Apache server** running on my home computer, I decided to put the "Unsafe" into it (of course this server is operating **purely locally**!), configured a GMC500+ counter to send its data to that server, implemented a PHP script which forwards the URL to GeigerLog, and programmed GeigerLog to handle this URL. To replicate, take these steps:

49 https://www.gmcmap.com/AutomaticallySubmitData.asp

50 http://httpd.apache.org/docs/2.4/de/mod/core.html see HttpProtocolOptions

Step 1: Enter “HttpProtocolOptions Unsafe” into the **apache2.conf** file and restart Apache.

Step 2: In the GMC500+ counter config set “Website” to the IP of the computer running GeigerLog (do NOT use “http://” before IP!), and make “Server URL” blank. (Easily done by using GeigerLog menu **Device → GMC Series → Set GMC Configuration …**).



Step 3: Copy the PHP script below into the Document Root of your Apache server (on Linux: /var/www/html/). Set $myIP to the above IP Address and $myPort to the port of GeigerLog’s server for WiFiClients (default=8000).

```php
<?php
$myIP = "10.0.0.20"; // Set your Forward-to IP
$myPort = "8000"; // Set your Forward-to Port
$myDevice = "GMC"; // Setting for GMC counter – don’t change!
$GeigerLog_URI = "http://" . $myIP . ":" . $myPort . "/" . $myDevice .
$_SERVER['REQUEST_URI'];

$ch = curl_init($GeigerLog_URI);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HEADER, FALSE);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>
```

Step 4: Run GeigerLog with config setting `WiFiClientType=GMC`. The figure shows a recent result running a GMC-500+ per USB-serial cable in 1 sec cycle (blue trace), and via WiFi with the fastest possible cycle of 1 min (brown trace). The result is the same apart from fewer data in the WiFi case.

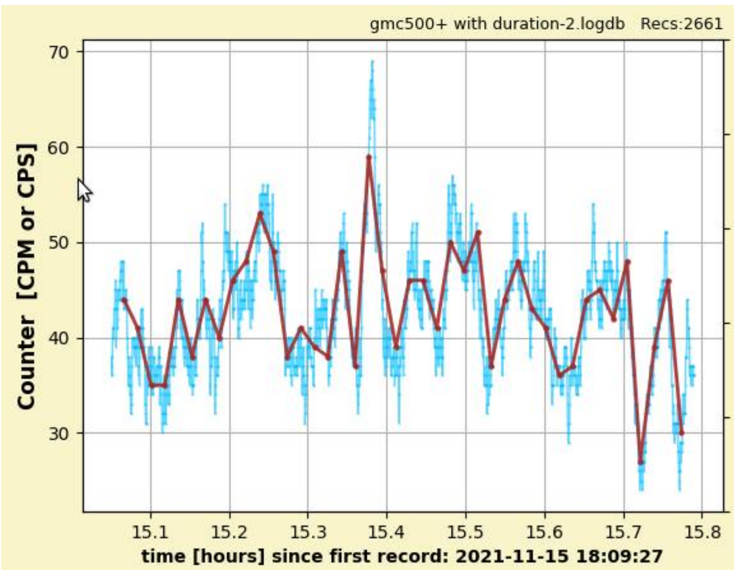


Figure 42: Running a GMC-500+ counter both via USB-Serial (blue) and WiFi (brown)

WiFiServer Devices

These are external devices which act as **Web Server** themselves and send data only when requested by GeigerLog. This is how most devices respond today, but now it is also wirelessly!

In Web terms GeigerLog has here the role of a browser and calls the WiFiServer device. In fact, you could type the same calls, which GeigerLog will issue, into the web address field of any browser (Firefox, Chrome, …) and the WiFiServer Devices will respond with the same answers.

GeigerLog has only three possible requests and expects responses in a certain format.

Let's assume the WiFiServer device has an IP address 'serverip', listens at Port number 'port', and offers the data at a subfolder 'folder' [51]) to the root directory. As an example we assume a WiFiServer Device providing Geiger counter CPM and CPS data, as well as Temperature, Pressure, and Humidity data from e.g. a BME280 sensor.

Request by GeigerLog	**Expected response to be given by WiFiServer device**	
http://serverip:port/folder/id		
	Format:	WiFiServerDeviceName
	Example:	AmbioMon++
http://serverip:port/folder/lastdata http://serverip:port/folder/lastavg		
	Format:	M, S, M1, S1, M2, S2, M3, S3, T, P, H, X [52])
	Example:	66, 1, , , , , , , 23.4, 998.1,6.5,

The ***/id** call serves the double purpose to verify that the device is present and getting its name.

The difference between ***/lastdata** and ***/lastavg** is that the former should always be the last available set of values, while the latter is an average over a certain period of time calculated by the WiFiServer device itself. This is of interest if e.g. the device samples in 1 sec intervals, but the reading is done slowly in 1 minute intervals. In this case an averaging of all data over 1 min by the WiFiServer device would be appropriate.

If the WiFiServer device cannot provide all values for all variables, then a simple comma must be inserted at the place of the missing variable; GeigerLog always expects 11 commas total.

Installation

In the `[WiFiServerDevice]` section of the GeigerLog configuration file `geigerlog.cfg` activate the **WiFiServer device**. You must set the IP address, Port number, and folder of the external WiFiServer device. Optionally set the other parameters timeout, tube sensitivity, data type, and variables.

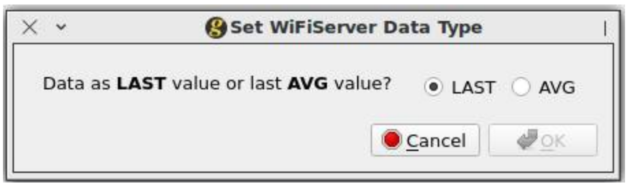
51 The folder name may be empty (folder=""); then the data will be served from the root directory.

52 Abbreviation M, S, M1, … are as introduced in Graph Dashboard – Visualize Your Recordings on page 23

Ping WiFiServer

Pinging a server allows to verify its presence and ability to respond to your requests. A success or failure note will be printed into NotePad.

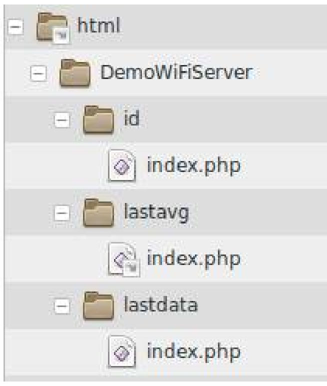
Set WiFiServer Data Type



Click menu **Device → WiFiServer Series → Set WiFiServer Data Type … .** In the dialogue you can select whether the **LAST** or **AVG** values will be used.

Note that the changes can only be applied when you are **not logging!** Otherwise the OK button is grayed out and inaccessible.

Testing



If you don't have such a device as hardware at hand, you can start by setting up a **web server with PHP**, using the file structure as shown in the picture. In this case the web server's document root is 'html'.

Naming all files 'index.php' in separate directories allows to call the directory only; no need to give a filename with php extension, which may get in the way when calling microprocessors!

Then use these simple scripts for getting it to behave like a real WiFiServer device:

For **id** save as file under `<web server document root>/folder/`**`id`**`/index.php:`

```
<?php
// Sends the ID of a WiFiServer Device in the form of a simple text string.
echo "WiFiServer Simulator on Apache PHP";
?>
```

For **lastdata** and same or similar for **lastavg**:

save as file under: `<web server document root>/folder/`**`lastdata`**`/index.php:`

save as file under: `<web server document root>/folder/`**`lastavg`**`/index.php:`

```
<?php
// Behaves as a WiFiServer Device yielding responses to /lastdata, /lastavg
// requests by returning a CSV string of data, like:
//    configured : CPM, CPS, Temp, Press, Humid, Xtra
//    response :   111.254, 1.197,,,,,,,21.000, 6.000, 48.000, 84.000
// Commas must be put in for missing values
//
// The data are synthetic, random data. The CPM/CPS data are approximately
```

```
// Poissonian, ambient data are random over meaningful intervals, see code.

function rand_pseudo_normal($av, $sd) {
/*
* generates an approximately normal distribution
* "Box–Muller transform" based random deviate generator.
* @ref https://en.wikipedia.org/wiki/Box%E2%80%93Muller_transform
*
* @param float|int $av Mean
* @param float|int $sd Standard deviation of Mean
* @return float
*/

$x = mt_rand() / mt_getrandmax();
$y = mt_rand() / mt_getrandmax();

return sqrt(-2 * log($x)) * cos(2 * pi() * $y) * $sd + $av;
}

function rand_pseudo_poisson($av){
// gets value from normal with StdDev = sqrt(mean)
// and takes abs to make all values positive.
// Reasonably ok as Poisson with $av >= 10

return abs(rand_pseudo_normal($av, sqrt($av)));
}

$CPM = rand_pseudo_poisson(100);
$CPS = rand_pseudo_poisson(1.67);
$T = rand( 20, 25);
$P = rand(980, 1030);
$H = rand( 40, 60);
$X = rand( 80, 100);

// sending: 'CPM, CPS, Temp, Press, Humid, Xtra'.
echo sprintf("%0.3f, %0.3f,,,,,,,%0.3f, %0.3f, %0.3f, %0.3f", $CPM, $CPS, $T, $P, $H, $X);
?>
```

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

• Clear NotePad	Delete all content from the NotePad
• Search NotePad	Search the NotePad for the occurrence of a text. Also conveniently accessible by CTRL-F!
• 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
• Geiger Tubes …	Set sensitivities 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: RadMon-Series	Functions related to RadMon Devices
• Submenu: AmbioMon-Series	Functions related to AmbioMon Devices
• Submenu: Gamma-Scout-Series	Functions related to Gamma-Scout devices
• Submenu: I2CSensors-Series	Functions related to I2C-Sensor based devices

- Submenu: LabJack-Series — Functions related to LabJack based devices
- Submenu: MiniMon-Series — Functions related to MiniMon based devices
- Submenu: Simul-Series — Functions related to Simul devices
- Submenu: Manu-Series — Functions related to Manu devices
- Submenu: WiFiClient-Series — Functions related to WiFiClient devices
- Submenu: WiFiServer-Series — Functions related to WiFiServer devices

Menu – Device – Submenu: GMC-Series

• Show Info	Prints some basic info about the device to the NotePad.
• 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.
• Set GMC Configuration	Opens a dialogue to set the counter's internal configuration.
• 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.
• 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.
• 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: RadMon-Series

• Show Info	Prints some basic info about the device to the NotePad.

- 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.
- Show Extended Info — Additional info on the device
- Configure — Configure settings of the AmbioMon device

Menu – Device – Submenu: Gamma-Scout-Series

- Show Info — Prints some basic info about the device to the NotePad.
- Show Extended Info — Additional info on the device
- Set Date + Time — Set the Gamma-Scout internal clock to computer time
- 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; required for history downloads and other operations.
- Set to Online Mode — The device allows logging to GeigerLog. The device’s log cycle can be set to 2, 10, 30, 60, 120, and 300 seconds.
- Reboot — Execute a reboot (Warm start) of the device.

Menu – Device – Submenu: I2CSensors-Series

- Show Info — Prints some basic info about the device to the NotePad.
- Show Extended Info — Additional info on the device
- Reset System — Reset the I2C ELV dongle and sensors

Menu – Device – Submenu: LabJack-Series

- Show Info — Prints some basic info about the device to the NotePad.
- Show Extended Info — Additional info on the device

Menu – Device – Submenu: MiniMon-Series

- Show Info — Prints some basic info about the device to the NotePad.

Menu – Device – Submenu: Simul-Series

- Show Info — Prints some basic info about the device to the NotePad.

- Set Properties — Opens a dialogue to set properties of the device

Menu – Device – Submenu: Manu-Series

- Show Info — Prints some basic info about the device to the NotePad.
- Enter Values Manually — Opens a dialogue to enter values manually

Menu – Device – Submenu: WiFiClient-Series

- Show Info — Prints some basic info about the device to the NotePad.

Menu – Device – Submenu: WiFiServer-Series

- Show Info — Prints some basic info about the device to the NotePad.
- Ping WiFiServer Device — Sends a “Ping” command to the device
- Set WiFiServer Data Type ... — Opens a dialogue to set data type Last or Avg

Menu – Log

Commands related to logging.

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

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

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

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

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

• Set Log Cycle	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 CSV file	Save all records from current log into a CSV file with extension 'csv'

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: AmbioMon Series	History functions related to AmbioMon Devices
• Submenu: Gamma-Scout Series	History functions related to Gamma-Scout Devices

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

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 AA Map — Show History Binary Data as a map highlighting the locations of bytes with AA value
- Show History Binary Data as FF Map — Show History Binary Data as a map highlighting the locations of bytes with FF value
- Save History Binary Data to File — Save the history binary data as a *.bin file, i.e. the binary data are extracted from the database and saved into a binary '.bin' file, compatible with earlier versions of GeigerLog.

Menu – History – Submenu: 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 – History – Submenu: Gamma-Scout Series

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

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

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

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

Menu – Web

- Show IP Status — Show GeigerLog's current IP and Port usage
- Set up Monitor Server … — Set up GeigerLog's Monitor server to allow monitoring GeigerLog from your Smartphone.

- Set up Radiation World Map … Set up the cycle to upload your data from a selected variable to the Radiation World Maps, see Radiation World Maps on page 103.

Menu – Tools

- 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 SuSt (Summary Statistics) of Plot Data Print summary statistics for all data currently shown in the plot to the NotePad
- Show Statistics of Plot Data In a pop-up window show detailed statistics for the data currently shown and selected in the plot
- 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 Poisson Test of Plot Data In a pop-up window show a 'Histogram with Poisson Fit' for the data currently shown and selected in the plot
- Show FFT & Autocorrelation of Plot Data In a pop-up window show a 'FFT & Autocorrelation' analysis of the data currently shown and selected in the plot
- Scaling … View and Edit the current value- and graph-scaling settings
- Save Graph to File Save the current graph as a PNG file
- Show Plot Data Show the Date&Time and variable values as currently shown in the plot
- Display Last Values Open a window showing all variables and their values
- Set Port 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 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 109 for more details.

Menu – Tools – Submenu: Autodiscover Port for Device:

- 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.

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.
- About GeigerLog A brief introduction to GeigerLog, as well as version and legal information

Toolbars

The individual toolbars **Main**, **Device**, **Log**, **History, Web, Tools,** and **Graph** are combined into a single toolbar, see 3. If preferred, they can be separated and relocated on the screen by grabbing their vertical bars on the left and moving them.

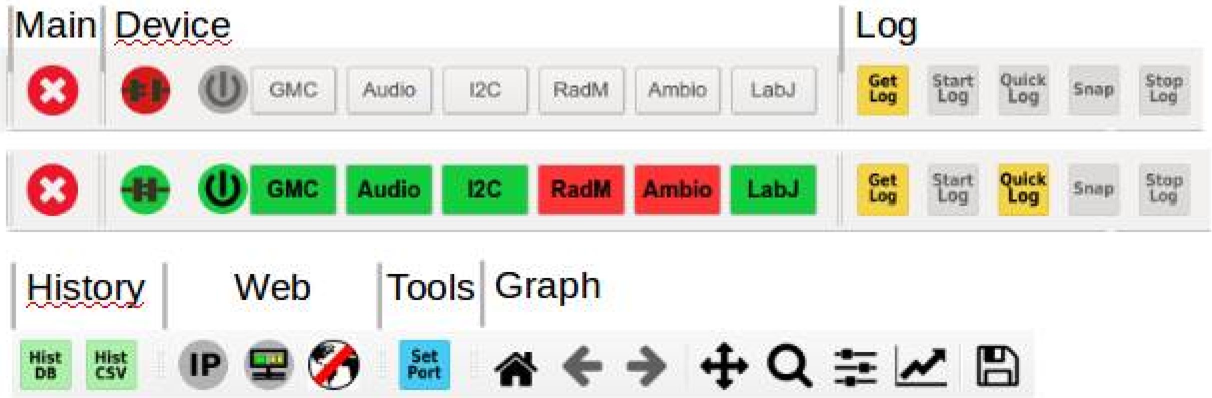


Figure 43: The toolbars

Upper-Top : non-connected status
Lower-Top : connected status
Bottom : remainder of toolbar

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

- Web - Show IP Status
  - Setup Monitor Server
  - Setup Radiation World Map, see Radiation World Maps on page 103.

- Set Port - A shortcut for the menu item **Tools → Set Port …**

- 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

Running GeigerLog on HiDPI Monitors

HiDPI monitors are displays with a pixel resolution higher than FullHd. FullHd typically means a pixel resolution of 1920 x 1080 (width x height), or “2k”, HiDPI monitors are typically labeled as “4k” or “8k”, meaning a width of approx. 4000 or 8000 pixels, resp. There is no fixed norm for either label, so 4k could mean e.g. 3840 x 2160, but also 4096 x 2304, and 8k could mean 7680 x 4320, or other “nearby” sizes.

The support of those HiDPI monitors by operating systems is still rudimentary and surely inconsistent, which is easily seen by the fact that some software output becomes too small to read, and requires to set “scaling” - if it exists. This triggers a different problem if part of the software supports scaling, and part not, where now the output may become too large to fit on the monitor.

GeigerLog allows to switch its adaptation to HiDPI on and off, and allows some tuning. You find the configuration file `geigerlog.cfg` in the section “Window”.

HiDPIactivation: this is set to “yes” by default. This should be ok for almost all situations, except that you must set this to “no” when GeigerLog is run on a Raspi 4 (status August 2020).

HiDPIscaleMPL: This compensates any HiDPI scaling for the graphs. For a FullHd monitor the 'scaling' set in the operating system is typically 100%. Any different scaling must be compensated.

Try it with the “auto” setting first. If this gives no good result, read the tuning instructions in the configuration file.

On an 8k monitor, with scaling set to 200%, HiDPI activated in GeigerLog, and the HiDPI-scaleMPL set to 50, the screenshot in Figure 44 shows the excellent result:

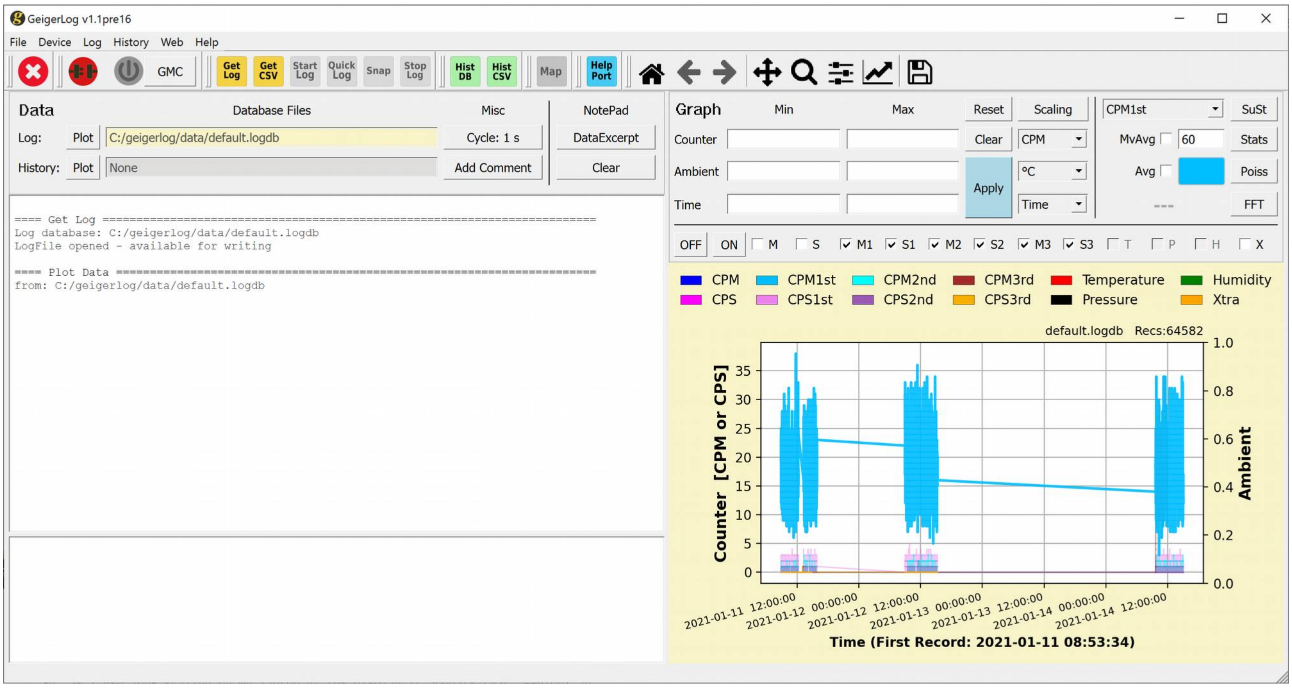


Figure 44: GeigerLog running on an 8k HiDPI monitor

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]

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.

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

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 only 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 often 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 **gtools** directory. None is required to run GeigerLog, but sometimes can be really helpful.

GeigerLog Pip Check – GLpipcheck.py

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.py
```

for output like this:

```
----------------------- GLpipcheck.py ------------------------------------------
GLpipcheck Version:  1.2
Python Executable:   /home/ullix/geigerlog/vgl310/bin/python3
Python Version:      3.10.0a6 (default, Mar  2 2021, 02:01:08) [GCC 5.4.0 20160609]

Listing of all Pip-found packages:
   Package             Version
   ------------------- ------------
   APScheduler         3.6.3
   cachetools          4.2.2
   ... lines removed ...
   SoundFile           0.10.3.post1
   terminaltables      3.1.0
   tornado             6.1
   tzlocal             3.0

GeigerLog REQUIRED packages and their version status
   pip                 21.3.1            OK
   setuptools          58.5.3            OK
   PyQt5               5.15.6            OK
   PyQt5-sip           12.9.0            OK
   numpy               1.21.3            OK
   scipy               1.7.2             OK
   matplotlib          3.4.3             OK
   sounddevice                           MISSING
   SoundFile           0.10.3.post1      OK

GeigerLog OPTIONAL packages and their version status
   pyserial            3.5               OK   REQUIRED for GMC, Gamma-Scout, I2C Series
   paho-mqtt           1.6.0             OK   REQUIRED for RadMon Series
   LabJackPython       2.0.4             OK   REQUIRED for LabJack Series
   pip-check           2.6               OK   Recommended Pip tool

Checking for updates ... Done

GeigerLog REQUIRED packages having upgrades available:
   Package             Version           Latest     Type
   numpy               1.21.3            1.21.4     wheel

GeigerLog OPTIONAL packages having upgrades available:
   Package             Version           Latest     Type
   paho-mqtt           1.6.0             1.6.1      sdist


Mini-HOWTO:
-----------------------------------------------------------------------------------------
for using 'pip':
NOTE:   when 'pip' itself or 'setuptools' can be upgraded, upgrade 'pip' first, then
        'setuptools', then any others
```

```
NOTE:   when using '=', '==', '>', '<' for version definition you need to use quotation
        marks around package designation
                    like: "anypackage == 1.2.3",  "anypackage >=1, <2"

Install and/or update package 'anypackage':
                    python3 -m pip install -U anypackage

Install and/or update multiple packages at once:
                    python3 -m pip install -U anypackage 2ndpackage 3rdpackage ...
        NOTE:
        If command fails, nothing at all may be installed!
        Continue with single-package installations.

Install package 'anypackage' in exactly version 1.2.3:
                    python3 -m pip install "anypackage == 1.2.3"

Install package 'anypackage' in version of at least 1 but not 2 or later:
                    python3 -m pip install "anypackage >=1, <2"

Remove  package 'anypackage':
                    python3 -m pip uninstall anypackage

Find    all available versions of package 'anypackage':
                    python3 -m pip --use-deprecated=legacy-resolver install anypackage==
```

GeigerLog Audio Check – GLaudiocheck.py

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

Start with:

```
python3 /path/to/GLaudiocheck.py
```

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):

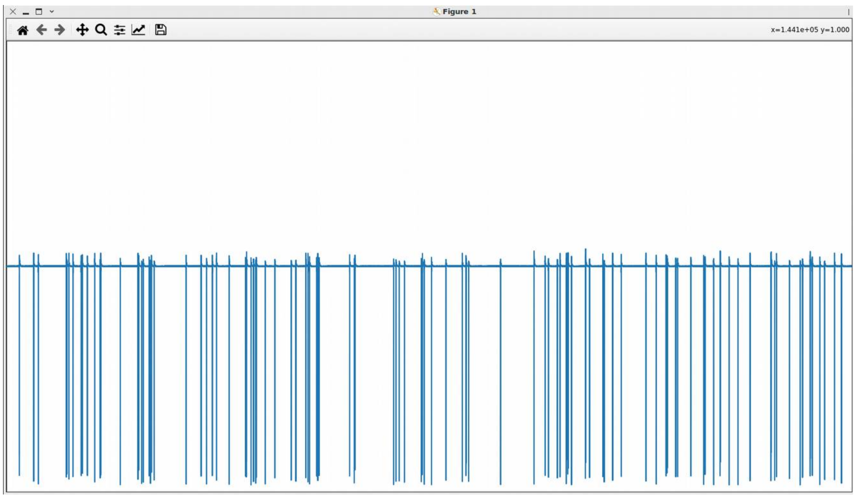


Figure 45: Output from the GLaudiocheck.py for an audio source with negative pulses

GeigerLog CSV File Merger – GLmerger.py

It allows to merge two GeigerLog created CSV files into one. It assures that the resulting CSV file has all variables set. Any records with the same DateTime stamp will be merged into a single record.

Start with:

```
python3 /path/to/Glmerger.py File#1 File#2 mergedfile
```

for output like this:

```
# File#1
#    Index,              DateTime, CPM,     CPS,     Temp,    Press,    Humid
         6, 2021-11-11 10:34:55,  91,     2.0,      3.0,      4.0,      5.0
         7, 2021-11-11 10:34:56, 106,     2.0,      3.0,      4.0,      5.0
         8, 2021-11-11 10:34:57,  99,     2.0,      3.0,      4.0,      5.0
         9, 2021-11-11 10:34:58, 108,     2.0,      3.0,      4.0,      5.0
        10, 2021-11-11 10:34:59,  92,     2.0,      3.0,      4.0,      5.0
        11, 2021-11-11 10:35:00,  99,     2.0,      3.0,      4.0,      5.0

#File#2
#    Index,              DateTime,  CPM1st,  CPS1st,  CPM2nd,  CPS2nd
         6, 2021-11-11 10:34:55,     43.0,      3.0,      0.0,      0.0
         7, 2021-11-11 10:34:56,     43.0,      0.0,      0.0,      0.0
         8, 2021-11-11 10:34:58,     42.0,      0.0,      0.0,      0.0
         9, 2021-11-11 10:34:58,     43.0,      1.0,      0.0,      0.0
        10, 2021-11-11 10:34:59,     44.0,      1.0,      0.0,      0.0
        11, 2021-11-11 10:35:00,     45.0,      2.0,      0.0,      0.0

#mergedfile
          DateTime,CPM,CPS,CPM1st,CPS1st,CPM2nd,CPS2nd,CPM3rd,CPS3rd,  T,    P,    H,   X
2021-11-11 10:34:55, 91,2.0,   43.0,    3.0,    0.0,    0.0,      ,      ,3.0, 4.0, 5.0,
2021-11-11 10:34:56,106,2.0,   43.0,    0.0,    0.0,    0.0,      ,      ,3.0, 4.0, 5.0,
2021-11-11 10:34:57, 99,2.0,       ,      ,      ,      ,      ,      ,3.0, 4.0, 5.0,
2021-11-11 10:34:58,108,2.0,   42.0,    0.0,    0.0,    0.0,      ,      ,3.0, 4.0, 5.0,
2021-11-11 10:34:59, 92,2.0,   43.0,    1.0,    0.0,    0.0,      ,      ,3.0, 4.0, 5.0,
2021-11-11 10:35:00, 99,2.0,   44.0,    1.0,    0.0,    0.0,      ,      ,3.0, 4.0, 5.0,
2021-11-11 10:35:01, 98,2.0,   45.0,    2.0,    0.0,    0.0,      ,      ,3.0, 4.0, 5.0,
```

You see that the 3rd row, marked in red, is the only one with a time stamp occurring only in the File#1.

Obviously, any variable must not be present in both files, or one would be lost in the merging process.

GeigerLog Serial Monitor - GLsermon.py

GLsermon.py allows to send commands and receive responses via any Serial Connection.

It supports any device, but it has very helpful extra support for the GMC counter series, which may need **binary coded** commands.

Start with:

to get Help:	`GLsermon.py -h`
to set Port and Baudrate:	`GLsermon.py -P <Port> -B <Baudrate>`
specific for **GMC** counter:	`GLsermon.py -P <Port> -B <Baudrate>` **`gmc`**

Stop with: `CTRL-C`

A <Port>	on Linux is:	**/dev/ttyUSBx**, x = 0, 1, 2, ...
	on Windows is:	**COMx**, x = 0, 1, 2, ...
The <Baudrate> is:		57600, 115200, …

In the examples below you see that the simple **getcps** command gets a 4-byte answer (not 2-byte!), and the quite complex **SPIR** command, requiring 5 binary coded bytes, can be send with ease.

A communication session with GLsermon.py connected to a GMC-500+ Geiger counter may look like this: (both input and output will be recorded to file `Glsermon.log`; you'll find it in the directory where GLsermon was started from.)

```
getver
>>>>> NOTE: Byte values printed to a Terminal may look incomplete! Compare
Length with expected length! <<<<<
Sending: '<GETVER>>'  (Length:9)

2021-11-08 18:30:19.775 :
Receive-Bytes:  15: GMC-500+Re 1.22
Receive-Values HEX:  47  4D  43  2D  35  30  30  2B  52  65  20  31  2E  32  32
Receive-Values DEC:  71  77  67  45  53  48  48  43  82 101  32  49  46  50  50
Receive-Values ASC:   G   M   C   -   5   0   0   +   R   e       1   .   2   2

getvolt
>>>>> NOTE: Byte values printed to a Terminal may look incomplete! Compare
Length with expected length! <<<<<
Sending: '<GETVOLT>>'  (Length:10)

2021-11-08 18:31:36.324 :
Receive-Bytes:   5: 4.9v
Receive-Values HEX:  34  2E  39  76  00
Receive-Values DEC:  52  46  57 118   0
```

Receive-Values ASC: 4   .   9   v

getcps

>>>>> NOTE: Byte values printed to a Terminal may look incomplete! Compare Length with expected length! <<<<<

Sending: '<GETCPS>>' (Length:9)

2021-11-08 18:32:38.104 :

Receive-Bytes: **4**:

Receive-Values HEX: 00 00 00 00

Receive-Values DEC: 0   0   0   0

Receive-Values ASC:

spir,0,0,0,1,0

>>>>> NOTE: Byte values printed to a Terminal may look incomplete! Compare Length with expected length! <<<<<

Sending: '<SPIR#>>' (Length:12)

2021-11-08 18:33:33.765 :

Receive-Bytes: 256: (Non-decodeable)

Receive-Values HEX:

55 AA 00 00 00 00 00 00 00 55 AA 01 55 AA 00 00 00 00 00 00 00 55 AA 01 55 AA 00 00 00 00 00 00
00 55 AA 01 01 05 00 00 00 02 00 01 00 02 00 01 00 00 00 00 02 00 00 00 00 00 00 00 01 00 00 00
00 00 02 00 00 00 01 01 00 00 01 00 00 00 01 00 00 00 00 02 00 01 00 01 00 01 01 01 01 01 01 00
00 00 00 00 00 00 01 01 00 00 00 00 00 01 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00
(more data; not shown)

Helpful External Software Tools

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

Editor VSCode

The Microsoft Visual Studio Code editor **vscode** is available on **Linux**, **Windows**, and **Mac**. In combination with a **Python extension** (https://code.visualstudio.com/docs/languages/python) it is a very powerful, professional editor for Python. Other extensions exist for other languages. Unfortunately, it has a bit of a steep learning curve.

This is the new development standard for GeigerLog.

Get it from https://code.visualstudio.com/.

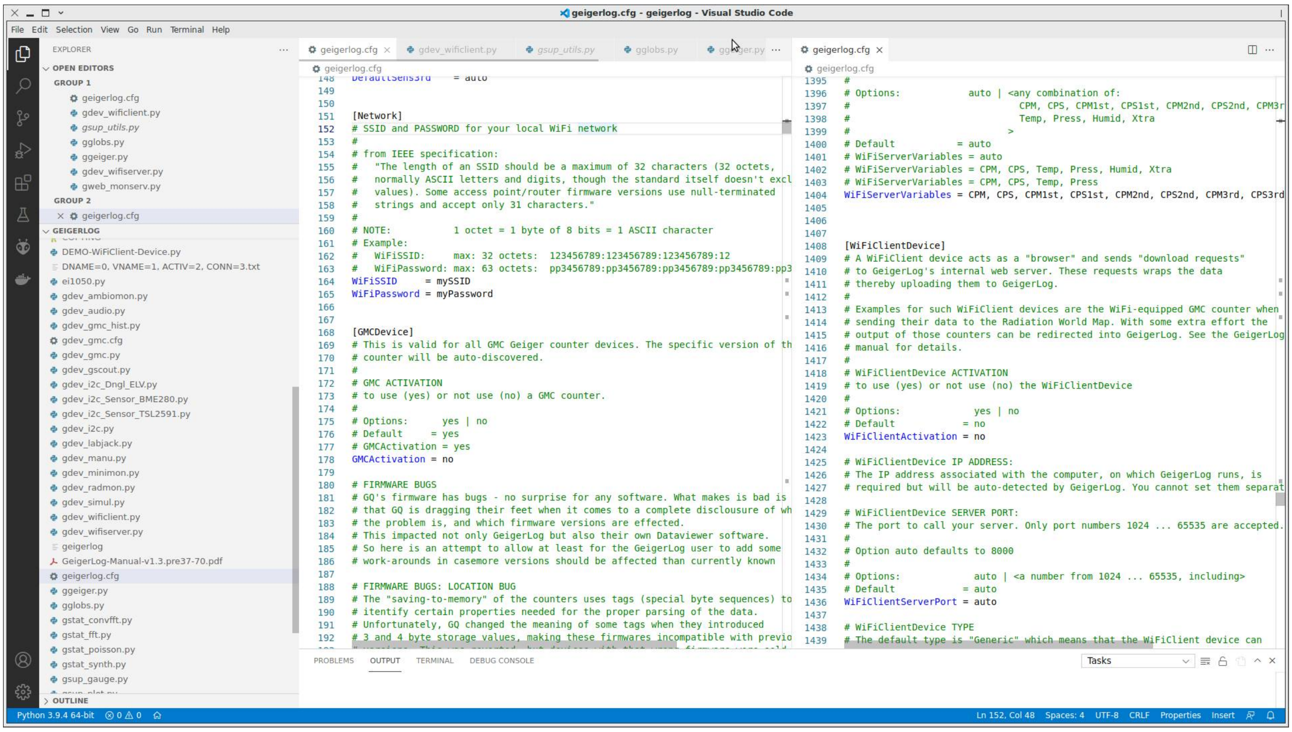


Figure 46: Editor Visual Studio code with file geigerlog.cfg opened in Split-Editor mode

Editor Geany

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

GeigerLog had in the past been developed on the editor **Geany**, which I do recommend as a general purpose editor.

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

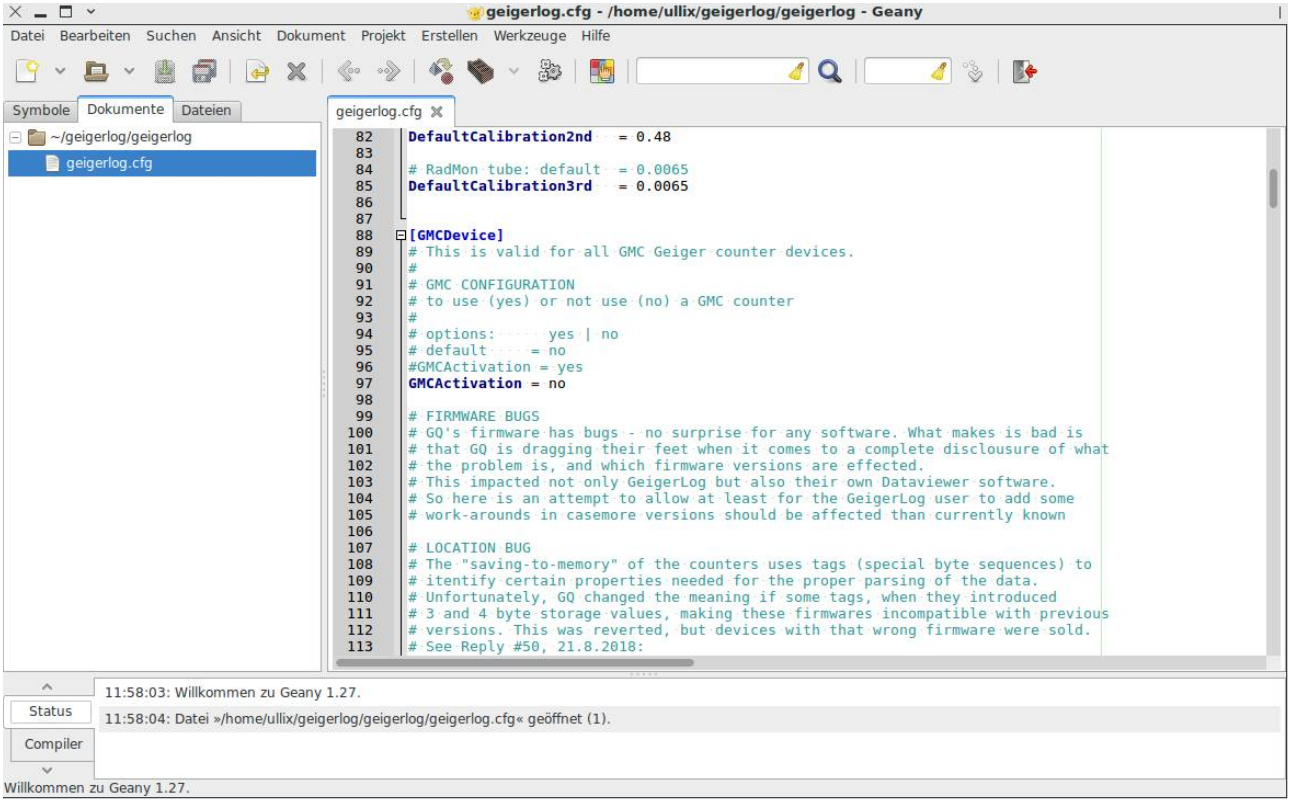


Figure 47: Editor Geany with file geigerlog.cfg opened

DB Browser for SQLite

As the Log and History files are 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:

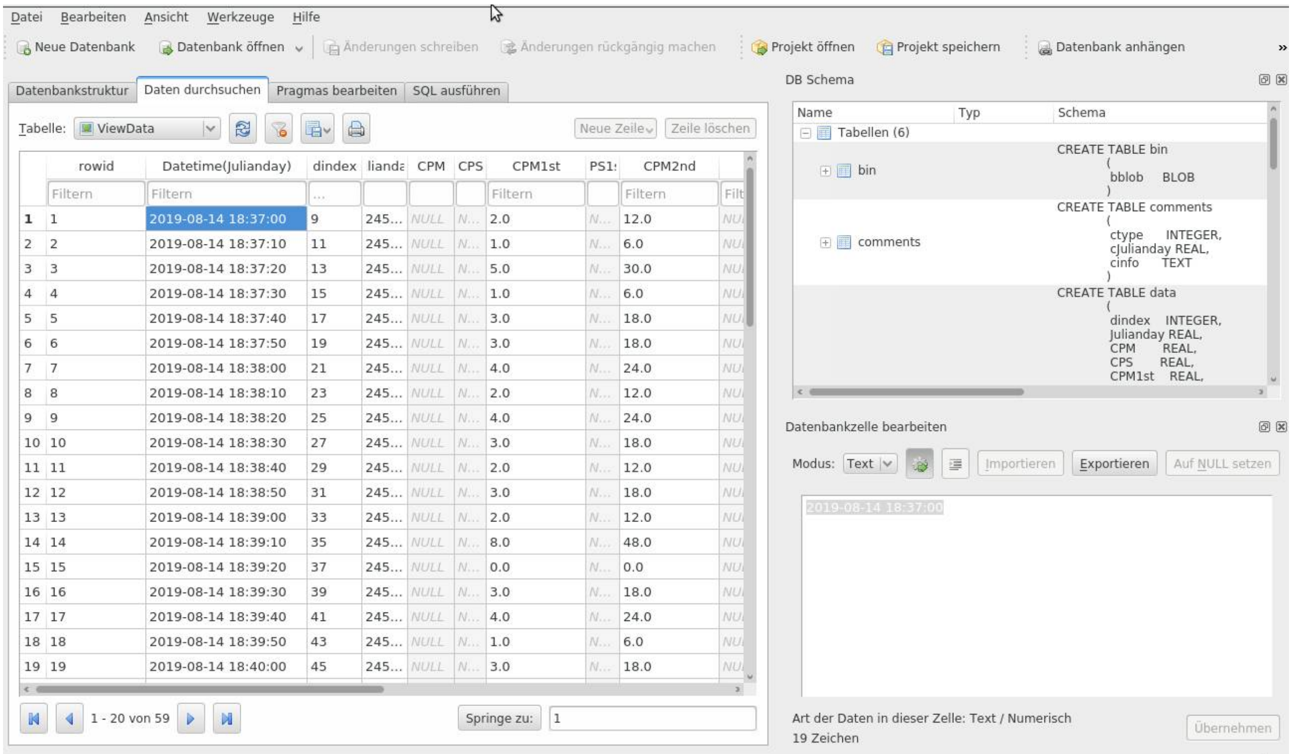


Figure 48: DB Browser for SQLite as tool to inspect Log files

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:

- gmcmap.com – This is the one supported by GQ Electronics
- radmon.org – Has well recovered from a hacking attack
- 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 WiFi equipped Geiger counter (series GMC-320+V5, GMC-500, GMC-600) to directly update their website. This is actually not such a good idea, see below.

Using GeigerLog to update the gmcmap.com site

You can support GQ's **GMCmap** using GeigerLog, and not only provide more meaningful data, but use any of their non-WiFi counters just as well. Actually, you can even use any other counter as long as it can be connected to GeigerLog.

This includes digitally connected counters, like e.g. RadMon, GammaScout, and analog-connected counters like the whole world of only-audio-click counters, including truly classic counters, the latest low-cost counters, or even the most modern solid state detectors using PIN photo-diodes [53]).

How to set this up is explained in chapter Set up Radiation World Map on page 29.

But there are several problems associated with the **GMCmap** site in particular.

Some words of caution

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

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

53 See chapter AudioCounter Devices on page 55, and Review Geiger Counter "Smart Geiger Pro (SGP-001)" from https://sourceforge.net/projects/geigerlog/files/Articles/GeigerLog-Review%20Smart%20Geiger%20Pro%20%28SGP-001%29-v.1.0.pdf/download

things like CPM=9999. But if you did something like that inadvertently you would discover that there does not seem to be a way to retract any such wrongly sent data.

Poor data will quickly destroy any value of those sites.

Instantaneous CPM: It is a bit more subtle, but diminishes the data quality nevertheless. GQ's potential upload is: CPM, Average-CPM, µSv/h reading. The latter two are optional. This lets me to conclude that CPM 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 [54]), 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 already is almost a 3fold difference!

GeigerLog will send CPM values always as averages, but allows the user to determine the length of the period, over which data are averaged. I suggest to have values collected for at least 30 minutes, more is better, before sending anything to the maps. Daily updates are quite appropriate!

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

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 a lot more extensive specifications.

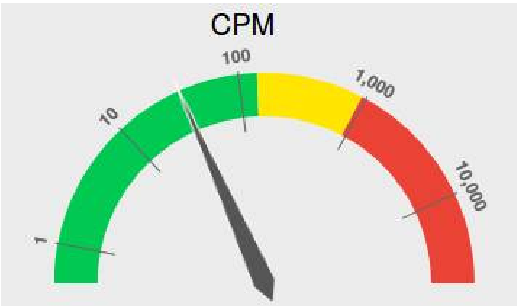
	Germany	USA	USA:Ger
Yearly exposure	20 mSv	50 mSv	2.5 x
equiv. to permanent:	2.3 µSv/h	5.7 µSv/h	
Lifelong exposure	400 mSv	2350 mSv	5.9 x
equiv. to permanent: (assuming in 40 years of work)	1.14 µSv/h	6.71 µSv/h	
	Grenzwerte	OSHA	

The differences are astonishing, to say the least! Over one year the US citizens are considered 2.5 times “tougher” than the Germans? And over the lifetime even 5.9 times tougher?

An even more intricate difference becomes visible when one computes dose rate per hour for yearly, lifelong and for each country: the German limit sinks twofold, while the US limit increases approx. 20%? Say what, the Germans become feebler as they age and need to be more protected, while the Americans grow even sturdier and take the extra?

This indicates a fundamentally different view between the two countries on how radiation effects health. I can’t imagine that this has anything to do with Science. It is a legal thing. Everything seems possible when facts don’t matter!

On what grounds do we set the radiation safety levels?



A gauge as shown on the right is such as easy, universally understood symbol for safety warnings with green for ok, yellow for caution, and red for danger.

But at what levels do we set the transition between the zones, from green to yellow, and from yellow to red?

With the legal situation of little help (see last chapter) from where do we take guidance?

The “natural background” could be considered a marker for a safe level – if it weren’t, mankind would not exist. The global average is 0.27 µSv/h [55]). The average dose of Finland is more than 3 times higher at 0.9 µSv/h. Even higher levels are found in the town of Ramsar (Iran), which has levels of 1.2 µSv/h average, at some locations in town even up to 15 µSv/h [56]). People live there as healthy as elsewhere. With this being a more than 50fold difference, it raises the question: which level is then “safe”?

The natural background results from the combined effects of radiation originating from the earth and from the cosmic radiation. The latter increases with altitude and at commercial aviation altitudes is much higher than the ground background. The cosmic radiation is lowest near the equator, and increases towards the poles. Thus your exposure depends on where you fly. Overall, the radiation exposure of air crews will be much higher than of other people, and the health of air crews could be an indicator for safe levels.

The US FAA has released a report in 1990 which said that cosmic radiation for domestic air travel is 6 µSv/h [57]). In a recent scientific publication [58]) an estimate was given for the average effective dose over the lifetime of aircrew at 5 … 6 µSv/h. So both estimates are close to the USA regulatory lifetime limit.

I decided for these values:

- Low = 0.9 µSv/h – transition green to yellow (Finland average)
- High = 6.0 µSv/h – transition yellow to red (Air crew and US regulatory)

These numbers nevertheless are arbitrary. They can be changed in the GeigerLog configuration file `geigerlog.cfg`, in segment DOSERATE THRESHOLDS.

These numbers are also exaggerated by the fact that they are for lifetime-long permanent exposures, but the situation for anyone experimenting with a Geiger counter is that there would at worst be only a short term higher dose exposure.

How to account for that, I don’t know. But I think there is plenty of leeway!

55 Numbers from https://en.wikipedia.org/wiki/Sievert unless otherwise stated

56 https://en.wikipedia.org/wiki/Ramsar,_Mazandaran

57 http://www2.ans.org/pubs/magazines/nn/pdfs/2000-1-3.pdf

58 From: https://academic.oup.com/jrr/article/59/suppl_2/ii1/4844965
Typical doses and dose rates in studies pertinent to radiation risk inference at low doses and low dose rates, Rühm et al., *Journal of Radiation Research*, Volume 59, Issue suppl_2, April 2018, Pages ii1–ii10, https://doi.org/10.1093/jrr/rrx093
Quote: **Air crew:** In Germany, for example, the average annual occupational effective dose to air crew was 1.9 mSv, in 2012 [28]. Maximum allowed flight hours for air crew are typically ~900 h per year, and some individuals accumulate up to 5 mSv per year or more. Assuming a maximum of ~40 working years, cumulative career doses of up to 200 mSv can therefore not be excluded. In such cases, mean effective dose rates of ~5–6 µSv/h can accumulate, which is consistent with typical dose rates at flight altitudes from secondary cosmic radiation of 2–7 µSv/h depending on altitude, latitude, and solar activity [29, 30].

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 135 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 94.

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

References

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

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

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

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

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

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

Appendix A – Look & Feel

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

They may differ between computers 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 [59]):

```
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 of the style name is required!

Fonts

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

59 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

NOTE: GeigerLog, beginning with its version 1.3.0, can auto-detect the presence of a serially connected device, and chose the proper settings. As long as hardware does not change, the GMC counters, the GammaScout counters, and the I2C devices should be **plug-and-play!**

If, however, anything fails, you will need to follow these instructions.

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

- GMC devices (Geiger counters from GQ)
- I2C devices (environmental sensors)
- GammaScout devices (Geiger counters from GammaScout)

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 [60]), while USB2 is nominal 480 MBit/s and USB3.2 reaches 20 000 MBit/s [61])!

It is even worse with the **Gamma-Scout** devices, which support only 0.01 MBit/s or less [62]). 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.

60 115 200 and 57 600 baud for GMC, up to 230 400 baud for the ELV dongle used for the I2c devices

61 https://www.elektronik-kompendium.de/sites/com/1310061.htm

62 9 600 and 2 400 (!) baud

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 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 [63]).

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 [64]); often a generic driver suffices.

Permissions:

Depending on circumstances, a different hurdle may exist for Linux, as a regular user (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

63 https://www.gamma-scout.com/wp-content/uploads/Gamma-Scout_Communication_Interface_V1.7.txt
64 http://www.gqelectronicsllc.com/comersus/store/download.asp

1) Set Port

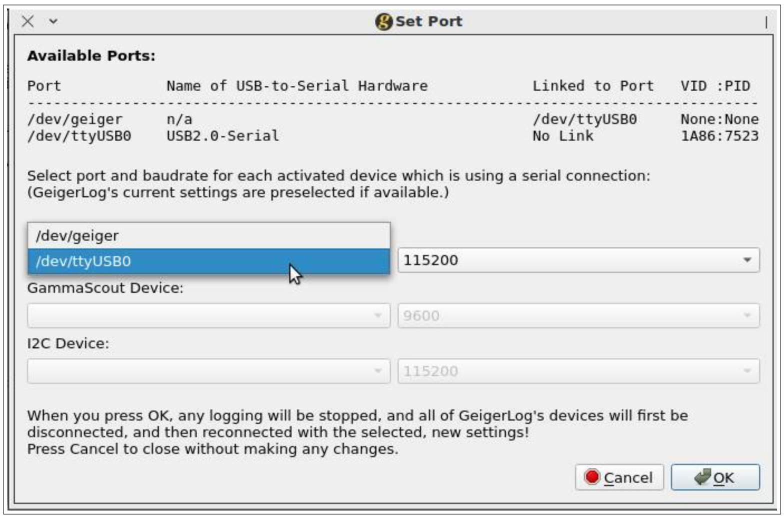


Figure 49: Show & Select Dialog for USB Port and Baudrate (result for a Linux system)

In the Tools menu of GeigerLog choose **Set Port …** or use the icon **Set Port.** A dialog pops up as shown in Figure 49. 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`.

2) Auto-Discovery

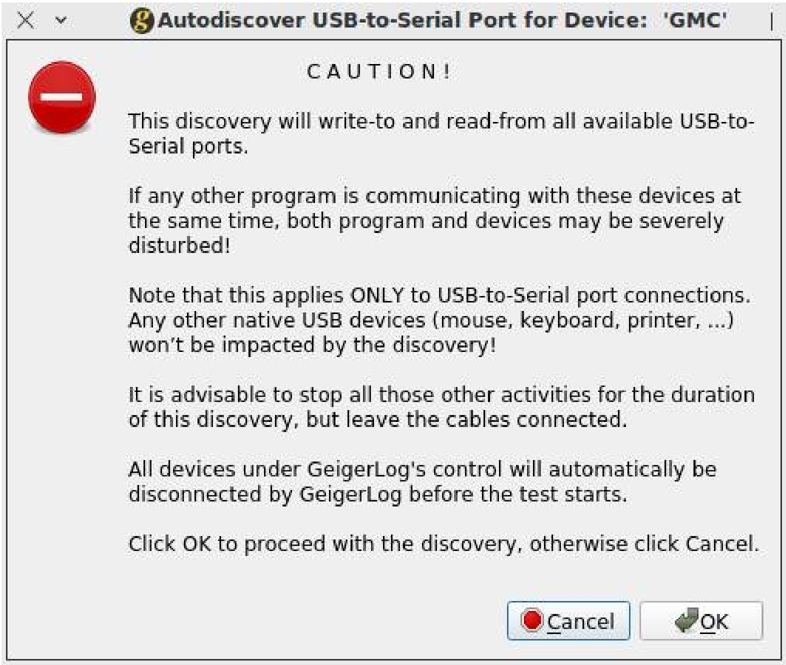


Figure 50: Autodiscovery USB-to-Serial Port Connection

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 **Tools → Autodiscover Port for Device: → <your device>** the warning shown in Figure 50 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 51 shown for a Linux system.

Figure 51: 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.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4447 . However, this list is not consistent with observed values at device 'GMC-300E Plus' with firmware 'GMC-300Re 4.20'. See here for even more differences: https://SourceForge.net/projects/gqgmc/files/gqgmc/GQ-GMC-ICD.odt/download

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

GQ has recently disclosed the configuration of the 500 and 600, but some details remain unclear so far. (see: http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4948), and at least 2 firmware bugs on the 500 series were discovered (see discussion in the topic). Please, report any problems via SourceForge (see Problems and Bugs on page 107).

The following list applies only to the 300 series:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

/
CFG_Save_DateTimeStamp, //this one uses 6 byte space

CFG_MaximumCFGBytes,

}EEPROMDATAT;

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

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

TOTAL_EEPROM_SAVE_TYPE

}HistoryDataFormatMarkingT;

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

```
SAVEOFF,

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

TotalSavedType

}SaveDataTypeT;
```

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

There is no official document from GQ on the storage format, except for some basic **“GQ Geiger Counter Communication Protocol”** from 2015 [65]) and a later update from 2018 [66]). But most credit for an initial description goes to **user Phil Gillaspy** [67]), who published it on his SourceForge site [68]).

Other info comes from my own 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.

---

65 https://www.gqelectronicsllc.com/download/GQ-RFC1201.txt

66 http://www.gqelectronicsllc.com/download/GQ-RFC1801.txt

67 https://SourceForge.net/projects/gqgmc/files/gqgmc/GQ-GMC-ICD.odt/download

68 https://sourceforge.net/projects/gqgmc/

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 GeigerLog CPM mask is effective.

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

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

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

Overflow

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

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

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

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

Page Boundaries

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

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

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

After the page is deleted, all bytes up to the end of the page are set to 255 (in gray). The former time fragments 19 (min) and 12 (sec) become regular counts (in white) and the remainder of the

Date&Time stamp beginning at 4098 (in yellow) has now become an ASCII tag with 85 bytes of supposed ASCII code following (only 3 bytes shown) [69].

```
 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
```

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 loose 1 second or minute, resp., in the time tag. But this is corrected with the next Date&Time stamp.

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

```
./geigerlog keepFF
```

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

Correcting a Wrong History

It is an annoying procedure. The following is suggested:

1. Download the full history from the counter, and look at the graph
2. Try to zoom into the critical zone with mouse-left-click and mouse-right-click followed by Apply. Do it until you are able to read the time and count value of a relevant data point
3. Search the *.his file for this data point and note the byte index (first column)
4. Search the *.lst file for this byte index, and determine which data need to be deleted
5. Using a program able to handle binary files, delete the segment just determined in *.bin file

69 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.

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.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4948

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

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

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

History Download issues

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

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

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

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

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

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

Configuration Issues

For the 300 series the configuration is stored in a memory of 256 bytes. There is confusion around the meaning of each entry (see Appendix D – The GMC Device Configuration Meanings, page 115), though most is understood.

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

Double-tube Counters

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

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

The tube choice will be auto-detected 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 plain 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=348!

On what grounds do you decide that the source is harmless or a severe threat?

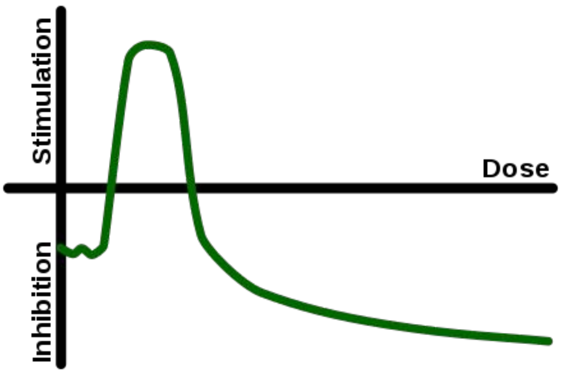
You probably would want first to equalize the readings from the counters. That is rather easy: you take a standardized radioactivity source, call one of the counters your reference, and determine correction factors for all others. So, all counters now give the same answer, but you still don’t know what 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**’ [70]), 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 µSv/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 µSv/h is implied! This assumption is anything but trivial, because we know e.g. from chemicals the effect called **hormesis** [71]), 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!

70 https://en.wikipedia.org/wiki/Sievert

71 https://en.wikipedia.org/wiki/Hormesis

For radioactivity the assumption of linearity is based on the LNT (Linear No-Threshold) [72]) 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** conversion factor independent of the rate.

GeigerLog's Method for the Conversion between CPM and µSv/h

GeigerLog uses the tube's '**Sensitivity**' as the single, constant calibration factor for the conversion of dose rate in CPM and dose rate in µSv/h. The sensitivity is given in units of **CPM / (µSv/h)**. This value tells you how many counts-per-minute a tube will generate in a gamma-radiating environment of 1 µSv/h. The more sensitive a tube is, the higher the sensitivity values will be!

Please note that this calibration factor is valid **ONLY** for gamma radiation!

Older data sheets may report the sensitivity in units of CPM/(mR/h)[73]), but as we are dealing with gamma radiation, we may use 1 mR/h = 10µSv/h.

The following table shows some tubes with known or claimed sensitivities, covering an range from 2 to 348 CPM/(µSv/h). Calibration Source is the source of radioactivity used to establish the sensitivity value.

Tube Name	Calibration Source	Sensitivity (CPM/ (µSv/h)	Comment
M4011	Unknown	154	Setting in GQ counters
J305	Unknown	154	Setting in GQ counters. This is another low-cost tube. It seems to be used by GQ as a – perhaps cheaper – replacement for the M4011 tube. Against all odds, the identical sensitivity setting is used!
SI3BG	Unknown	5.15	Setting in GQ counters (2nd tube in GMC-500+)
SI3BG	Th & K	2.08	own experiments using Th and K. May have beta component! http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=5369
SI3BG	Synchrotron	2.38	Synchrotron data provided by GeigerLog user Ikerrg; sensitivity relative to M4011=154
SBM20	Co-60	132	Russian Spec Sheet: http://www.gstube.com/data/2398/ "Gamma Sensitivity Co^{60} (cps/mR/hr) = 22"

72 https://en.wikipedia.org/wiki/Linear_no-threshold_model

73 One often finds units given as: "CPM/mR/hr". Apart from the non-standard use of 'hr' for 'h', this is even mathematically wrong. Correct would be: "CPM/mR * hr" or the setting of parentheses: "CPM/(mR/h)".

SBM20	Ra-226	174	Russian Spec Sheet: http://www.gstube.com/data/2398/ "Gamma Sensitivity Ra^{226} (cps/mR/hr) = 29"
LND712	Co-60	108	LND spec sheet: "GAMMA SENSITIVITY CO60 (CPS/mR/HR) = 18 " https://www.lndinc.com/products/geiger-mueller-tubes/712/
LND7317	Co-60	348	LND spec sheet: "GAMMA SENSITIVITY CO60 (CPS/mR/HR) = 58" https://www.lndinc.com/products/geiger-mueller-tubes/7317/
LND7317	Unknown	379	Setting in GQ's GMC-600 counters, see user Kaban, Reply#3 http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4948
LND7317	Cs-137	334	Geiger counter: "Radiation Alert Ranger": "Sensitivity (cpm per mR/h) = 3340" https://seintl.com/products/radiation-alert-ranger
SGP-001	Unknown	12	Not a Geiger tube, but a solid state gamma radiation sensor using PIN photo-diodes; can be used as a GeigerLog AudioCounter http://allsmartlab.com/eng/294-2/

The conversion between dose rates in CPM and in µSv/h is done with these formulas:

Dose Rate [CPM] = Dose Rate [µSv/h] * Sensitivity [CPM / (µSv/h)]
Dose Rate [µSv/h] = Dose Rate [CPM] / Sensitivity [CPM / (µSv/h)]

Note: This use of 'Sensitivity' constitutes a difference between GeigerLog Version 1.0 and later and the previous versions, which used 'Calibration Factor' for the conversion instead! But the difference is simple: Sensitivity is the inverse of the Calibration Factor.

The reason for the change is that the quantity Sensitivity is the standard by which established tube manufacturers report this property of their tubes, and is easier to grasp.

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 sensitivities of 154, 2.08, 379, 108, 11.1, and 2.38?
> (The 2 sets of numbers have the exact same meaning, of course!)

The Dilemma

A sensitivity 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 sensitivity for the M4011 tube in a GMC Geiger counter? Well, we don't know!

To my knowledge, GQ has never reported how the sensitivity 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 sensitivity 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^{226} (cps/mR/hr)	**29**
Gamma Sensitivity Co^{60} (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:

Sensitivity	**new-style**	**old-style (calibration factor)**
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 sensitivity:	**154 CPM / (µSv / h)**	0.0065 µSv / h / CPM

GQ's sensitivity 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 sensitivity. 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 [74]).

As shown in Figure 52 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.

74 http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4571

When measured with the radioactive sources Potassium and Thorium, the SBM20 shows elevated counts of 41% … 45% for both sources, as shown in Figure 53.

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 not only gamma, but also strong beta emitters.

Therefore from the experiment shown in Figure 53 we can **NOT** draw the conclusion that the SBM20 is more sensitive **to gamma** than the M4011; the question remains open.

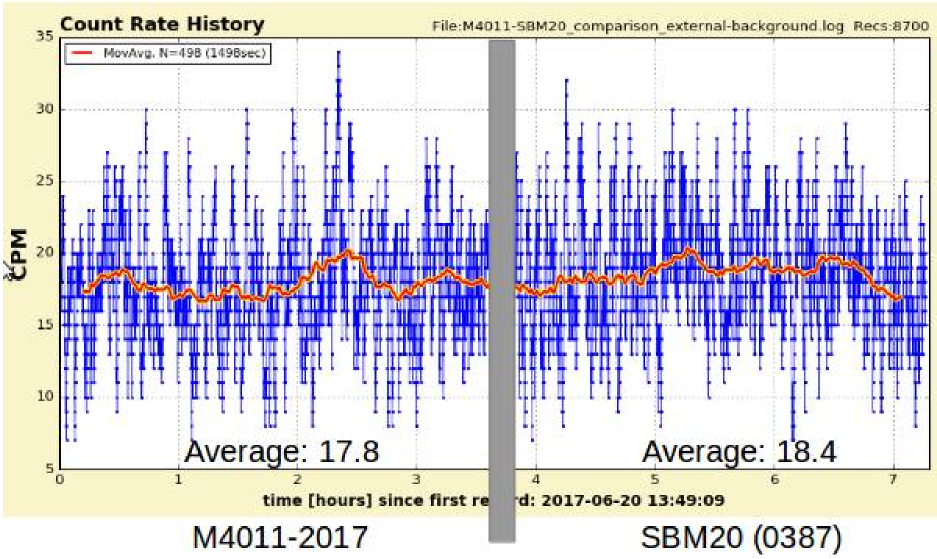


Figure 52: Comparison of a M4011 and SBM20 tube
The tubes are inserted into a GMC-300E+ counter and measure background

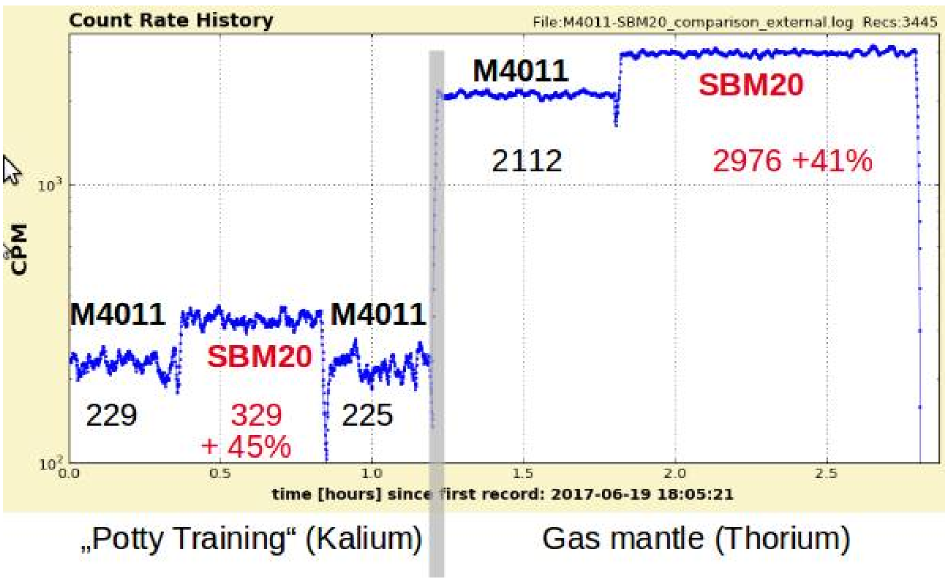


Figure 53: Comparison of a M4011 and SBM20 tube
The tubes are inserted in a GMC-300E+ counter measuring a Potassium and Thorium source, resp.

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 sensitivity is for beta!

And likewise, using a sensitivity with unknown specification, and likely relevant for gamma only, on tubes with sensitivity for alpha when measuring alpha radiation, is simply additional nonsense! When you measure any alpha radiation, report the CPM and describe tube used and setup. But do not report dose rate in µSv/h, because that tells the reader that you did not understand what you were doing!

The Energy Dependence of the Sensitivity

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.

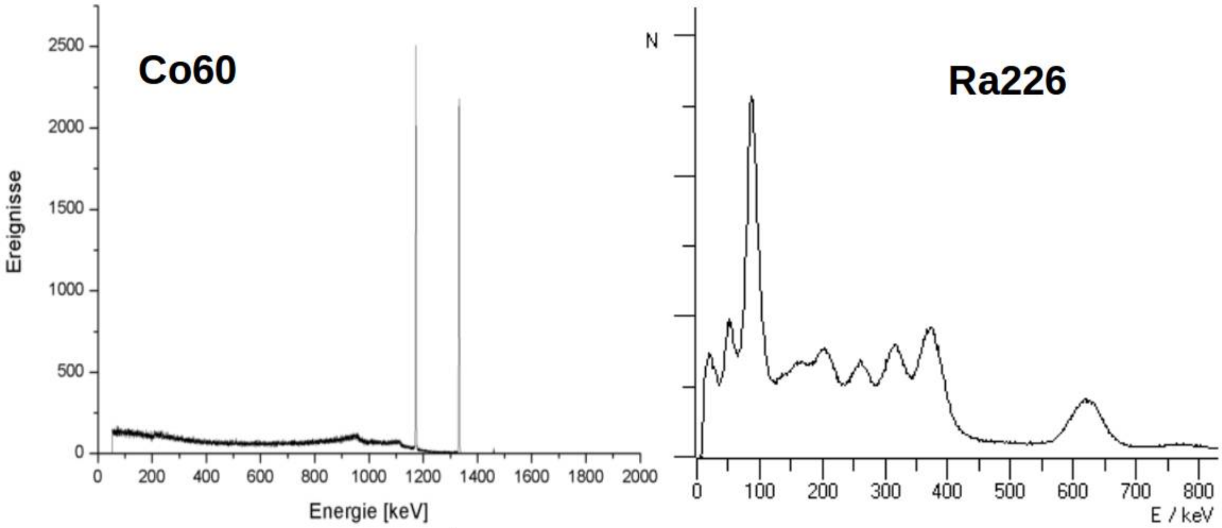


Fig. G1 Gamma Spectra of Ra226 and Co60

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. One example is found here [75]):

75 The N-16 Gamma Radiation Response of Geiger-Mueller Tubes, https://hps.org/hpspublications/articles/allard.html

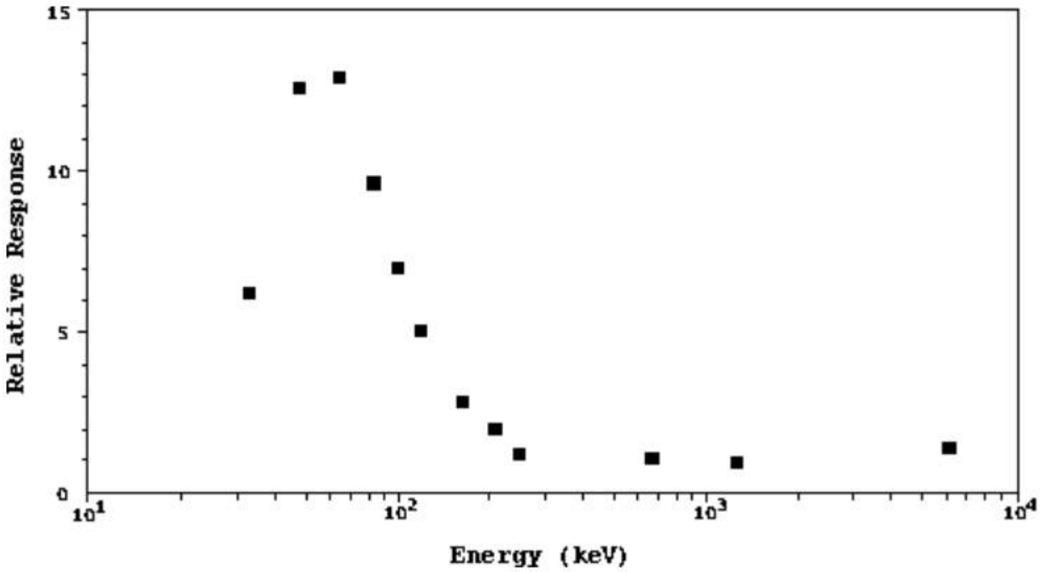


Figure 54: Relative Response of a Geiger Tube with Respect to Gamma Energy

Photon Energy Response of N116-1 miniature "thin wall" GM tube, normalized to unity at 662 keV with beam perpendicular to detector wall.

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!

Summary

1. The sensitivity 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 sensitivity is **NOT** valid for a beta emitter
4. The sensitivity 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 Co60, 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 Error: Reference source not found on page Error: Reference source not found. A broader discussion can also be found in this GQ forum topic: https://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=5357; in particular look into Reply #9 and Reply #33.

But, this discusses a mood 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: [76]).

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

GeigerLog now uses a default calibration of 154 CPM / (µSv/h) (old value: 0.0065 µSv/h / CPM) for the M4011 tube, and 379 CPM / (µSv/h) (old value: 0.002637 µSv/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.

Sensitivity 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 [77]). This is a device which can also generate strong gamma radiation.

---

76 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.

77 https://en.wikipedia.org/wiki/Synchrotron

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 55.

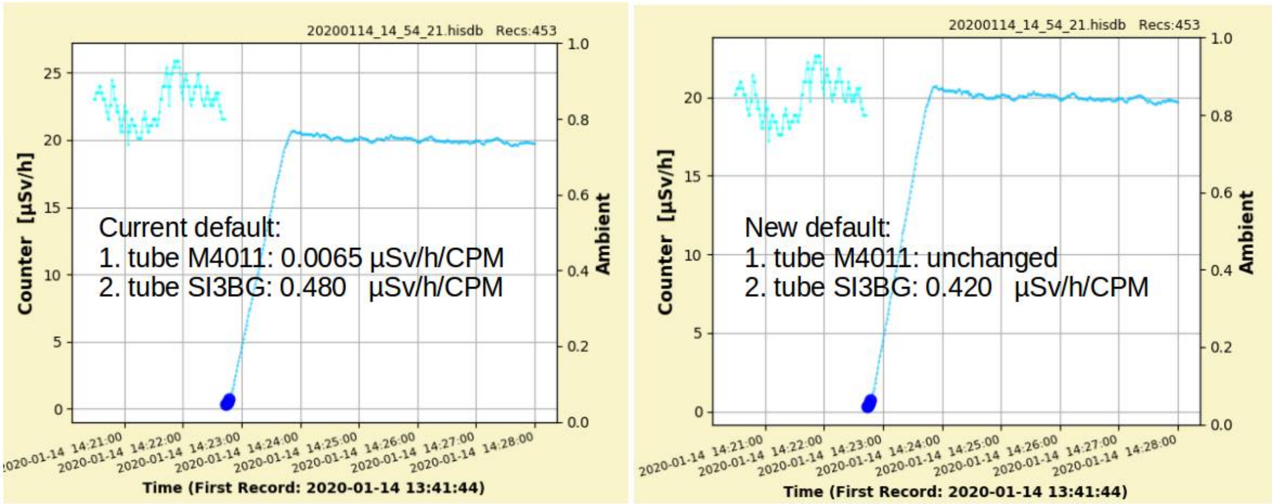


Figure 55: Synchrotron radiation captured with a GMC-500+ counter

History read out and analyzed with GeigerLog

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

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 2nd tube is 154 / 2.08 … 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. If these conditions are met, GeigerLog will run on **Linux**, **Windows**, **Mac**, and **other systems**!

Python – the proper version is Python 3.10

GeigerLog 1.3 requires **Python version 3.6 or later**. It was confirmed to work on Python 3.6, 3.7, 3.8, 3.9, and 3.10, both on Linux and on Windows 10.

Python version 3.5 and earlier are no longer supported by GeigerLog. Those Python versions are past their end-of-life anyway.

The latest is currently (Dec 2021) version **3.10**. This is now the preferred version.

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** [78]) 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 be for all users, and has to be installed with administrative rights. However, the Pip installed packages will be installed per user; so administrative rights will NOT be needed!

Verify the current Python installation status on your machine

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

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

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

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

The responses will tell you which commands are valid and which one starts which version of Python.

We will now assume that 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.

Verify the current Pip installation status on your machine

Additionally, you’ll need to make sure you have pip available. You can check this by running:

```
python3 -m pip --version
```

If `pip` isn’t already installed, then first try to bootstrap it from the standard library:

```
python3 -m ensurepip --default-pip
```

see: https://packaging.python.org/tutorials/installing-packages/

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.

Reminder:

The GeigerLog package contains the very convenient tool `GLpipcheck.py`.

Find it in the **gtools** directory, and start with:

`gtools/GLpipcheck.py`

It will list all installed packages, all needed by GeigerLog which are installed with their versions, all required by GeigerLog but missing, and all required by GeigerLog, for which an update is available. See GeigerLog Pip Check – GLpipcheck.py on page 95.

Verified to work with GeigerLog 1.3.0 under Python 3.10.0, latest versions as of January 2022:

- pip latest version: 21.3.1
- setuptools latest version: 60.5.0
- PyQt5 latest version: 5.15.6

• PyQt5-sip	latest version: 12.9.0
• matplotlib	latest version: 3.5.1
• numpy	latest version: 1.22.1
• scipy	latest version: 1.7.3
• pyserial	latest version: 3.5
• paho-mqtt	latest version: 1.6.1
• sounddevice	latest version: 0.4.4
• SoundFile	latest version: 0.10.3.post1
• LabJackPython	latest version: 2.0.4

Not required by GeigerLog, but recommended for installation:

• pip-check	latest version: 2.7

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 chapter Appendix I – Advanced Use of Pip on page 146, 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 --use-deprecated=legacy-resolver install numpy==
```

What if all installed well but GeigerLog fails to run?

To get more information on the problem let GeigerLog help you: 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 have been 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.

Still not found the problem?

Start GeigerLog with the options debug, verbose, very verbose, 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/ .

Linux – Installation

This was tested with **Ubuntu Mate 16.04 LTS, 18.04 LTS, 19.04, 20.04 LTS**.

Installation of Python and Pip

Install Python Version 3.10. 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 3.10 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 details see Other Packages on page 136)

```
gtools/GLpipcheck.py
```

Installation of GeigerLog

Copy the `geigerlog-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 [79]).

This also allows to configure variants of a specific Python version, e.g. using different Pip installed modules. E.g., I use Python versions 3.5, 3.6, 3.7, 3.8, 3.9, and 3.10 simultaneously on the same computer in such a virtual environment, each with their own set of pip modules, and they remain completely separated.

79 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 Ver 3.10. Download from: https://www.python.org/downloads/windows/.

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

https://www.python.org/ftp/python/3.10.2/python-3.10.2-amd64.exe

Open it **with administrative rights!**

In the installer:

- check: add Python 3.10 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\Python39’
  - 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 installation status on your machine on page 135) 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 details see Other Packages on page 136)

```
gtools/GLpipcheck.py
```

Installation of GeigerLog

It is suggested to place GeigerLog directly under `c:\`. Unzip the content of `geigerlog-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 command window, but in addition always also into the program log file `geigerlog.proglog` in the `data` directory of GeigerLog.

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

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

Note the '**w**'! 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 3.10. 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 installation status on your machine on page 135.

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 details see Other Packages on page 136)

```
gtools/GLpipcheck.py
```

Installation of GeigerLog

Copy the `geigerlog-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 Installation

I tested this on a **Raspi4** using its default Raspian installation based on **Buster**. 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:

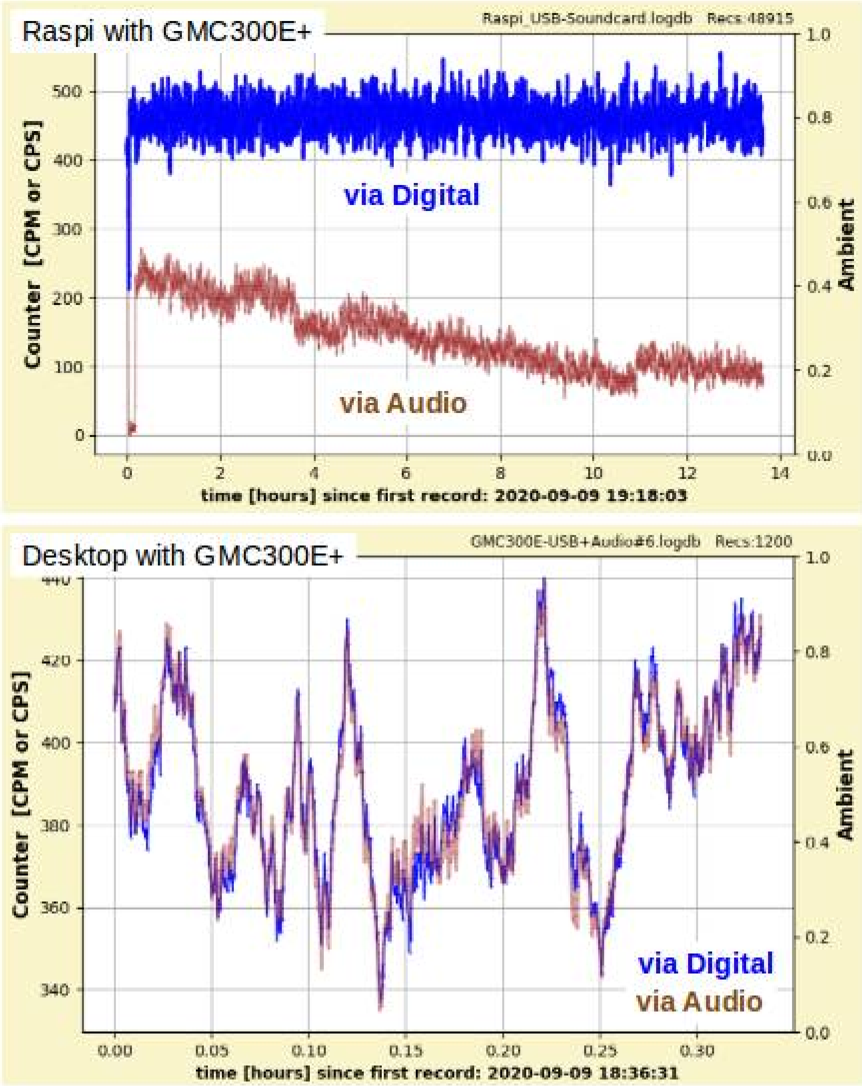


Figure 56: Comparing Raspberry Pi and Desktop Computer

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 tools and commands are:

Tool: GLpipcheck.py

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

```
python3 /path/to/GLpipcheck.py
```

See an example output here: GeigerLog Pip Check – GLpipcheck.py on page 95.

Tool: Pip-check

Pip-check is also based on Pip and 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 2.7 (Jan 2022).

Install or update pip-check:

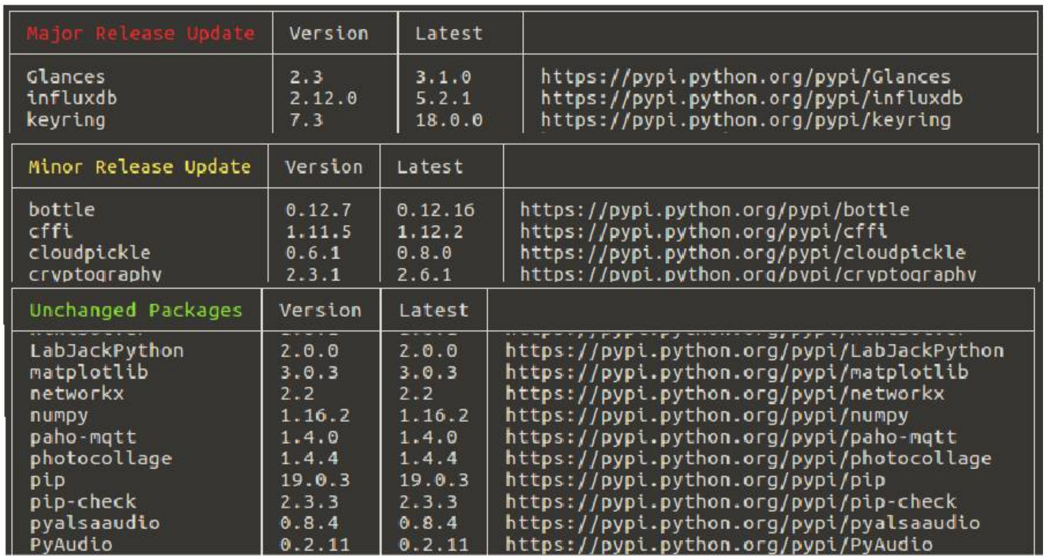
```
python3 -m pip install -U pip-check
```

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	https://pypi.python.org/pypi/Glances
influxdb	2.12.0	5.2.1	https://pypi.python.org/pypi/influxdb
keyring	7.3	18.0.0	https://pypi.python.org/pypi/keyring
Minor Release Update	**Version**	**Latest**	
bottle	0.12.7	0.12.16	https://pypi.python.org/pypi/bottle
cffi	1.11.5	1.12.2	https://pypi.python.org/pypi/cffi
cloudpickle	0.6.1	0.8.0	https://pypi.python.org/pypi/cloudpickle
cryptography	2.3.1	2.6.1	https://pypi.python.org/pypi/cryptography
Unchanged Packages	**Version**	**Latest**	
LabJackPython	2.0.0	2.0.0	https://pypi.python.org/pypi/LabJackPython
matplotlib	3.0.3	3.0.3	https://pypi.python.org/pypi/matplotlib
networkx	2.2	2.2	https://pypi.python.org/pypi/networkx
numpy	1.16.2	1.16.2	https://pypi.python.org/pypi/numpy
paho-mqtt	1.4.0	1.4.0	https://pypi.python.org/pypi/paho-mqtt
photocollage	1.4.4	1.4.4	https://pypi.python.org/pypi/photocollage
pip	19.0.3	19.0.3	https://pypi.python.org/pypi/pip
pip-check	2.3.3	2.3.3	https://pypi.python.org/pypi/pip-check
pyalsaaudio	0.8.4	0.8.4	https://pypi.python.org/pypi/pyalsaaudio
PyAudio	0.2.11	0.2.11	https://pypi.python.org/pypi/PyAudio

Figure 57: Output of pip-check in a Terminal

Command: List all versions for which an update is available:

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

Command: 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, cycler, numpy, python-dateutil, py-
parsing
Required-by:
```

Command: 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
```

Command: 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: 1.1, 1.2, 1.2.1, 1.2.2, 1.2.3)
ERROR: No matching distribution found for paho-mqtt==
```

thereby giving you the available versions.

Note: Unfortunately, this nice trick is no longer working. However, this workaround is available:

```
python -m pip --use-deprecated=legacy-resolver install paho-mqtt==
```

This legacy-workaround might be removed in the future, but as there had been so many complaints by users, an alternative will (hopefully) be available then!

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: