GeigerLog Manual

Version 0.9.04
June 2017

Author	ullix
Credits	Phil Gillaspy for extended documentation of Geiger counter commands GQ Electronics LLC for documentation
Copyright	Copyright 2016
License	GPL

Introduction

GeigerLog is a Python based program for use with GQ Electronic's [1]) GMC-3xx and GMC-5xx line of Geiger counters. It was developed on Linux Ubuntu Mate 16.04 but should also work on Windows and Mac as long as a Python 2.7 with PyQt4 environment is available.

The program allows reading of Geiger counter data, printing to screen and logging to file. Comments can be added to the log file during logging. The history stored on the Geiger counter can be extracted and converted into files that can be printed and analyzed.

Also, from all data - Log data and History data - graphs can be created and shown as Count Rate History versus time. The Count Rate can be shown in units of CPM or CPS or µSv/h. 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. Both graph axes can be in either fixed scale or auto-scaled mode. The graphs can be stretched, shifted, and zoomed for details, and saved as pictures in various formats (png, jpg, tif, svg, ...).

Time ranges can be set to plot data within that range and to analyze statistical properties of those data. For further statistical analysis a histogram of the data with a Poisson Distribution fit to the data can be plotted, and a frequency analysis by Fast Fourier Transform (FFT) can be done.

The ranges can be entered manually or by left and right mouse clicks for left (min) and right (max) range limits.

During logging the graph is live auto-updated. All manipulations of plots and data ranges can be done during ongoing logging without disturbing it.

Some Geiger counter functions can be controlled from GeigerLog, and changes made without interrupting logging.

The USB port used and its baud rate for the connection with the device can be auto-discovered, and used directly to connect to the device.

All communication with the device is checked for errors, which unfortunately do occur occasionally. The program attempts to auto-recover from an error, and continues if successful, which it is in most cases.

A genuine recording of Geiger counter data from an international flight is included, indicating count rate increase with altitude, and reduction of cosmic rays when going from north towards the equator.

Availability

The most recent version of GeigerLog, including this manual, can be found at project GeigerLog at Sourceforge: https://sourceforge.net/projects/geigerlog/, and is also included within the packages.

Other uses

While GeigerLog was developed specifically for the described purpose, it can be adapted with minor modifications for other types of data loggers. E.g. it is currently in use for long time recording of temperature and humidity using the Labjack brand of data loggers.

---

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

Installing and Starting GeigerLog

The software was developed on Python 2.7 and comes in two packages: one is just the **scripts** (and associated resources like icons), the other is a so-called **bundle**, containing in addition also the run-time environment. This latter All-in-One package has no further installation requirements.

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

The scripts

The scripts are only a few hundred kB big and require that you have a Python 2.7 environment on your computer, together with all required modules. All these modules are available in the repositories of Linux Ubuntu Mate 16.04.02. See Appendix I on page 31 for details on required packages.

If these software needs are met, GeigerLog will also run on Mac and Windows, and other systems.

The bundle

The bundle is several 10 MB big and contains the scripts, the resources **and** the environment to run them. You don‘t have to install a separate Python environment and all its modules. However, there is no harm done when those exist; they can coexist without impact on each other!

This bundling is done with PyInstaller. The Python scripts are now “semi-compiled”; you can’t edit them any more. The bundles are specific to the systems, on which they are created, i.e. a Linux bundle can only be run on Linux, and a Mac bundle only on Mac. Currently only a Linux bundle exists.

Installing and starting the scripts

1. The file is named ‘geigerlog-scripts-vXYZ.zip’ (xyz is the version number).
2. Make sure you have Python 2.7 installed, including all modules called in the scripts.
3. Start GeigerLog with

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

Installing and starting the bundle

1. The file is named ‘geigerlog-bundle-vXYZ.zip’ (xyz is the version number).
2. In order to run the GeigerLog in the bundle case, you **MUST** change into the geigerlog directory first; it will fail otherwise! In the terminal do:

```
cd /path/to/geigerlog
./geigerlog
```

   When you start from the file manager, first double-click the geigerlog **directory** to make it your working directory, then double-click the GeigerLog **program file** to start it!

Look & Feel

The Python software – both the scripts and the bundle – depend on the host computer for the Look & Feel. If GeigerLog does not look the way you like it, see Appendix A.

The GeigerLog Window

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

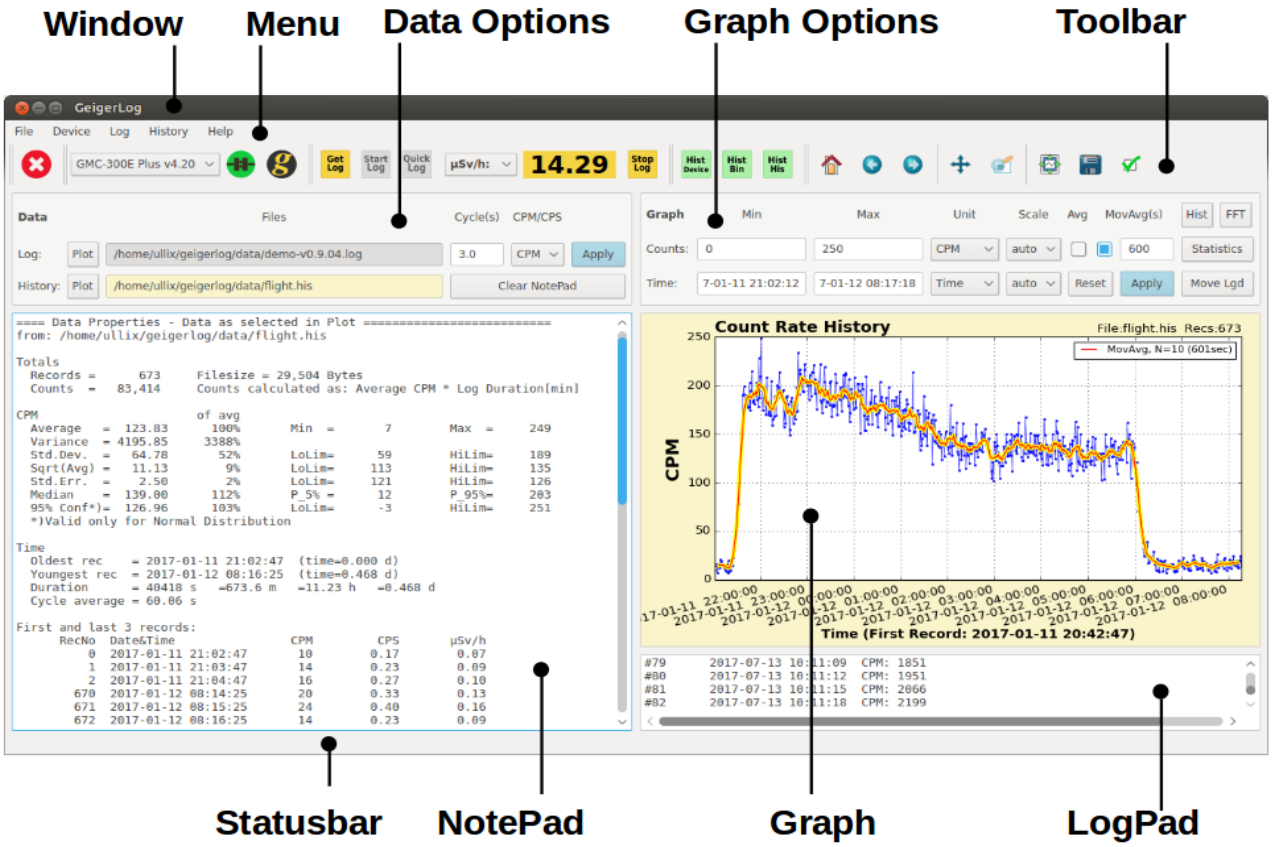


Fig. 1 GeigerLog Window with Annotations

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

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

Data Options: Info on, and plotting of, loaded Log and/or History files, and settings for logging.

Graph Options: Settings to configure your graph.

Toolbar: A toolbar with icons for quick mouse access to the more frequent actions. During logging the last value read is shown in big numbers in units of either CPM, CPS, µSv/h, mSv/d, or mSv/a.

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

NotePad: A scratch-pad type of area to hold various textual and numeric information. Clear the NotePad with the 'Clear NotePad' button located in the Data Options area.

Graph: Graphs will be shown here

LogPad: Once logging has started you find here the most recent log entries

Introducing GeigerLog – a Quick Demo

GeigerLog can be run without a connected Geiger counter device in order to analyze **existing** data. This is best shown with a quick demo, which will guide you through typical usage steps:

With GeigerLog running, click on menu ‘History’ → ‘Get History from Binary File’ and select **flight.bin**. The original data from an international flight from Germany to the Maldives will be loaded and displayed as a graph, showing the Count Rate History as CPM versus Time-of-Day.

In the Graph Options, click the drop-down button currently showing ‘CPM’, and select ‘µSv/h’. Click the Apply button. 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’.

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

Click the check button under ‘MovAvg(s)’ and Apply. A Moving Average is shown as an overlaid red-yellow line. But the currently used 60 seconds are inappropriate for this data set; enter ‘600’ and Apply again. Better. Since the data were collected by the Geiger counter in the ‘CPM, Saving every minute’ mode, 600 seconds means the data are averaged over 10 minutes, equal to 10 data points.

Click the ‘Move Lgd’ button repeatedly to move the legend in the graph to a place, where it is less disturbing to the visual.

Click the ‘Clear NotePad’ button in the Data Options on the left side. Click the ‘Statistics’ button in the Graph Options. Some statistics of the currently shown data of the flight data are printed into the NotePad. In the File menu click ‘Print File Stats’, and the statistics for the whole flight will be printed.

Click ‘Reset’, then ‘hours’ under Time Unit, then left-click on the vertical line near 10 h, and Apply. Then click button ‘**Hist**’, and a ‘Count Rate 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 second icon from the right to save the current graph as an image file. The availability of image formats depends on the host 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!

Running GeigerLog with Your Geiger Counters

Two operating modes are supported by GeigerLog: Logging and History. But first you have to establish a connection between Geiger Counter and computer.

Connecting the two might be more challenging than expected, but give it a try first: Connect the USB cable, start GeigerLog, and select menu **Device → Connect Device**.

- If successful, a printout into the NotePad **in black** will confirm the connection. You are set to continue.
- If unsuccessful, a printout **in red** will tell you the reason. You will be advised to run **USB Autodiscovery** from menu **Help**. If you are lucky, a dialog box will pop-up and tell you that it found a Geiger Counter and let you click the OK button to use it right away.

  However, you might want to make this more permanent by editing the configuration file geigerlog.cfg in section **Serial Port** to make these just found settings the default.

  For more complicated situations see Appendix B on page 20.

It is now assumed that a successful connection of the computer with the Geiger counter has been made, and GeigerLog has switched its connection icon from

disconnected  to connected  . Note that you can easily toggle the connection state by clicking this icon on the toolbar!

For a computer connection with the 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 can download the history.

But for all new radiation measurements the Geiger counter must be powered on. This power switching can be done manually directly at the device, or easier from GeigerLog (menu Device → Switch Device ON). GeigerLog's device power icon will change its state from

Power OFF  to Power ON  . Note that you can easily toggle the power state by clicking this icon on the toolbar!

Selecting Your Geiger Counter Model

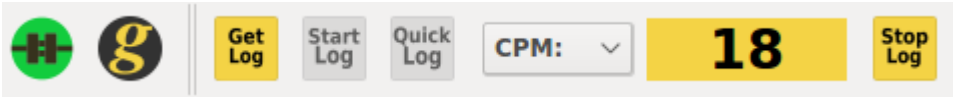
The toolbar has a selector for the Geiger counter models. Basically GeigerLog works the same for all devices except for some workarounds for the different firmware, firmware bugs, and memory sizes and more. Always choose your device model! See Appendix F on page 28 for details.

The default setting assumes a GMC300E+ with firmware v4.20. If your model is not listeer, choose the closest one. If it does not work properly, please see Bugs and Problems in this manual.

You can redefine the default model by modifying the configuration file geigerlog.cfg in section **Device**.

Logging

GeigerLog does control the logging. Once a Log file has been loaded and the device is connected and it is powered on you can start by clicking the **Start Log** icon in the toolbar. GeigerLog sends a command to the Geiger counter, asking for a new value, and reads the new value. It will immediately be stored in the Log file, be plotted in the graph, be printed into the LogPad, and be shown in the toolbar in your selected unit of either CPM, CPS or µSv/h:



This request is repeated after the cycle time has expired. This cycle time is set in the **Data Options** area. There you also set whether the default Counts per Minute (CPM) or Counts per Second (CPS) will be used. After any changes are made here, click the Apply button and the new settings will become effective immediately.

The QuickLog button is helpful if you want to just see current counts and don't care about storing the data. It automatically uses the default.log file to collect data. Note that this file is overwritten every time you click the QuickLog button! If you want to attach data to a previous QuickLog recording, choose StartLog instead.

A note on the logging cycle

The Geiger counter needs only 1 ms (millisecond; probably less, perhaps even as little as 0.1 ms) to 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.

Likewise, when CPM is selected, the counts during the last minute are summed up. Hence you get the complete picture 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; at least 30 sec, and then use the Moving Average (see Graph Options) for further smoothing the data.

The program GeigerLog could sample far more frequently – it is restricted in the GeigerLog code to a 0.1 second cycle, but could go up to 10x faster, even with plotting. However, while this may be relevant for other applications, it is not for Geiger counter activities.

History

The 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).

The original GeigerLog protocol for the download has now been modified, as a 5 min download is really inconvenient: the download will now be stopped when 8192 bytes of hex value FF has been read. This is the 'empty' value on 2 subsequent pages of 4k bytes each.

However, if the memory overflows the ringbuffer (see Appendix E) storage principle becomes effective, and the memory is overwritten beginning at the bottom. In this situation the whole memory is filled with data, and there will never be 2 pages of empty values. Hence the whole memory will be read! If you don't need the content of the memory, I suggest to erase it every once in a while.

But as the firmware remains undisclosed by GQ, it is possible that something unexpected may happen.

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

See Menu – History for the commands to download the History from device or file.

Menus and Commands of the Menus

Menu – File

Commands to plot data, statistics, histogram and exit the program

- Plot Log — Plot the log file data (if loaded)
- Plot History — Plot the history file data (if loaded)
- Print File Stats — Print statistics for the whole file currently shown in the graph into the NotePad area
- Print Plot Stats — Print statistics for only the currently plotted part of the graphed file into the NotePad area
- Show File Histogram — In a pop-up window show a histogram ‘Count Rate Histogram with Poisson Fit’ for data of the whole file
- Show Plot Histogram — In a pop-up window show a histogram ‘Count Rate Histogram with Poisson Fit’ for those data currently shown in the plot
- Show File FFT — In a pop-up window show a FFT analysis of the Count Rate data of the whole file
- Show Plot FFT — In a pop-up window show a a FFT analysis of the Count Rate data of those data currently shown in the plot
- Exit — Exit the program

Menu – Device

Commands related to the Geiger counter device, its status, configuration, operating mode.

- Connect Device — Connect computer with device.
- Disconnect Device — Disconnect computer from device.
- Show Device Info — Prints some basic info about the device on the NotePad; see also Appendix D for some content included in info.
- Show Device Configuration — Prints the device configuration as binary data in human readable format to the NotePad. Also see Appendix D.
- Switch Power ON — Switches the device power ON (as if pressing the Power button on the device).
- Switch Power OFF — Switches the device power OFF (as if pressing the Power button on the device).
- Switch Alarm ON — Switches the device alarm ON.

- Switch Power OFF — Switches the device alarm OFF.
- Switch Speaker ON — Switches the device speaker ON.
- Switch Speaker OFF — Switches the device speaker OFF.
- Switch Saving Mode of Device — Switches the device's mode of saving to the internal memory. Can be:
  - CPS, save every second
  - CPM, save every minute
  - CPM, save hourly average
  - OFF (no history saving)
- Set Date + Time of Device — Synchronizes computer and device time by setting the device's date and time to the computer time.
- Reboot Device — Reboots the device.
- FACTORYRESET Device — Does a factory reset. Your customization is lost, and the internal memory is cleared.

Menu – Log

Commands related to logging, i.e. the computer calls the Geiger counter to send a value, reads it, stores it in a file, and displays it numerically and graphically.

- Get Log File 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.
  If you select an existing file, new data will be **appended** to this file!

  After loading a file will always be plotted if it contains data, which can be plotted
- 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 value will immediately be shown in the graph, in the LogPad below the graph, and in the toolbar in the selected unit.
- Stop Logging — Stops logging; the last value remains visible in the toolbar but is grayed out.
- Quick Log — Start logging using the file default.log. The file will be emptied, before logging starts. If you want to continue logging into a previously selected default.log file, then choose Start Logging instead.

- Add Comment to Log — Adds a comment to the log file; does not disturb logging or graphing.
- Show Log Data — Prints the log data into the NotePad.
- Show Log Comments — Print only the comment lines of the log into the NotePad
- Show Log Data as Excerpt — Prints only the first and last few lines of the log, helpful for quick inspection.

Menu – History

Commands related to downloading the history stored on the internal memory of the Geiger counter.

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

  GeigerLog reads the data from the internal memory of the Geiger counter, and stores an exact copy as a binary file with the extension '.bin'.

  It also stores two derived files. One gets the extension '.lst', and provides a list of the binary data in a human readable form (helpful for review).

  The other gets the extension '.his' and contains a log of the count rates, extracted by GeigerLog parsing the binary history data. This file will be plotted. It has the same format as a log file. You could rename this file to the extension '.log' and then could append data to it by logging; but this rarely ever makes sense.

  If you want to forward a data file to someone else or archive it, it suffices to use the '.bin' file. The other derived files can always be created from it.
- Get History from Binary File — When the Geiger counter memory has already been downloaded into a binary file, you can use this file to produce the derived files '.lst' and '.his', and plot the data
- Get History from Parsed File — When a '.his' file already exists, you can use this file to plot the data. No attempts are made to recreate the '.bin' and '.lst' files.
- Show History Binary Data — Print the binary data from the '.lst' file to the NotePad

- Show History Parsed Data — Prints the data from the ‘.his’ file to the NotePad
- Show History Tags/Comments — Prints only those lines from the ‘.his’ file 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 Geiger counter via its Main Menu → Save Data → Note/Location.
- Show History Data Excerpt — Prints the first and last few lines of both the ‘.lst’ file and the ‘.his’ file to the NotePad

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
- Options — The options and commands available for starting GeigerLog from the command line, see chapter Starting GeigerLog with Options
- System Info — Useful information for development and debugging
- USB Autodiscovery — GeigerLog makes an attempt to determine your port and baud rate automatically, and shows the report in a pop-up dialog box. If a single Geiger Counter was found, you can take the found settings for this session and are given advice how to make this permanent. Otherwise info is given on how to proceed. See Appendix B for more details.
- About GeigerLog — A brief introduction to GeigerLog, as well as version and legal information

Toolbar

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

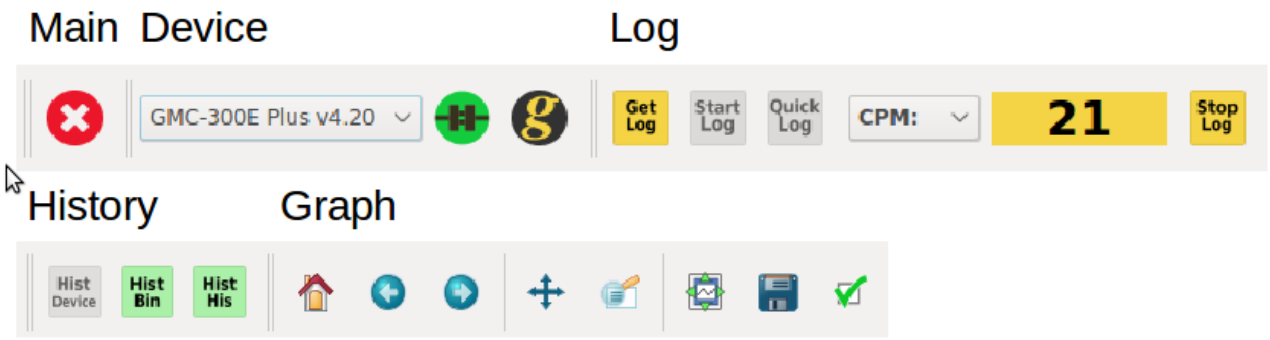


Fig. 2 The toolbars

- Main - Exit GeigerLog
- Device - Geiger Counter Selector (see Appendix F)
  - Toggle Connection
  - Toggle Power of Geiger Counter
- Log - Get a Log File
  - Start Logging
  - Start a Quick Log
  - Selector for different count rate units
  - Count rate in the unit to its left
  - Stop Logging
- History - Get History from device
  - Get History from binary file
  - Get History from parsed file
- 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

Data Options

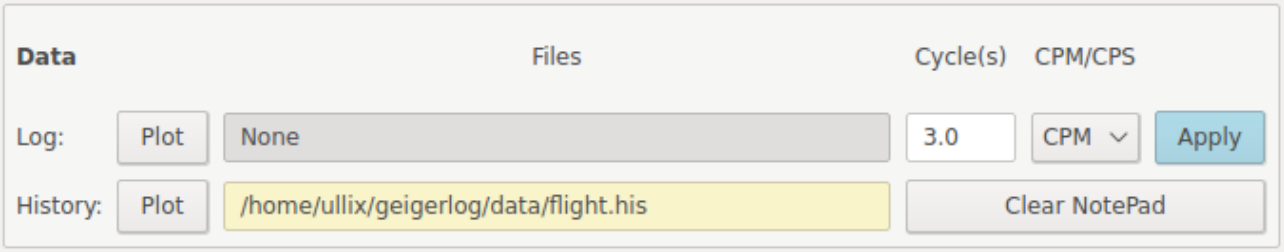


Fig. 3 Data Options

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

The **Cycle(s)** defines the cycle time in seconds between two requests by the computer during logging, and the drop down selector CPM/CPS defines whether Counts per Minute (CPM) or Counts per Second (CPS) will be requested from the counter. Changes become effective immediately with pressing the Apply button. Changes can be made at any time, also during ongoing logging.

The Clear NotePad button clears the NotePad area below.

Graph Options

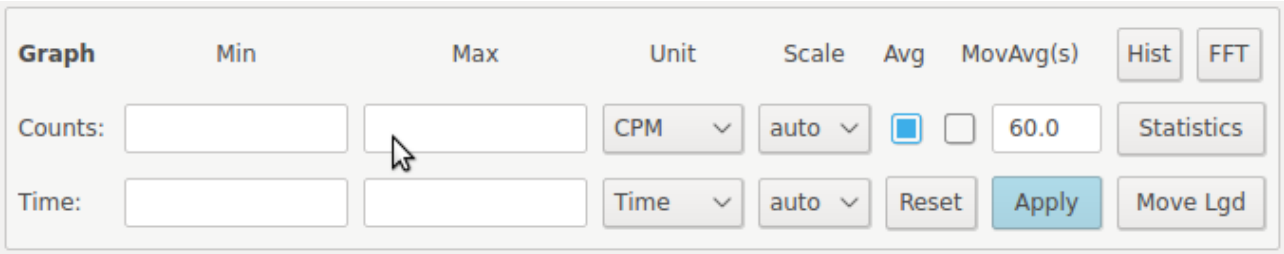


Fig. 4 Graph Options

The graph is laid out as counts on the vertical axis versus time on the horizontal axis.

The counts axis can have the units selectable by the drop-down button as CPM, CPS or µSv/h. The latter uses the calibration factor as built into the firmware of CPM * 0.0065 = µSv/h [2]). This setting can be changed in section **Device** of the configuration file geigerlog.cfg.

If the scale is set to auto, GeigerLog will determine the min and max of the count axis; to set your own limits change scale from auto to fix.

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

If the checkbox ‘Avg’ is checked then a horizontal black line will be drawn at the average value of all plotted data, and this average value ± Standard Deviation will be shown in the graph’s legend.

2 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 on page 29.

If the data can be approximated by a Normal Distribution, two horizontal dashed black lines will be drawn indicating the theoretical 95% range for the plotted data set. If GeigerLog determines that the condition of Normal Distribution is NOT met, then no 95% range lines will be drawn.

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/

A moving average [3]) (MovAvg) will be plotted as a red-yellow line, if the checkbox is checked. Enter a duration in seconds over which the MovAvg is to be calculated. The average will be plotted using a number of data points determined from the average cycle time of the record. If the cycle time had been changed during the recording, this may not be adequate; adjust the duration entered to achieve a better fit.

Always press the Apply button to make the changes effective.

The 'Move Lgd' button allows to move the graph's legend to a position less disturbing to the visual.

The 'Statistics' button will print statistics on the plotted data into the NotePad area. (Printing the statistics of either the whole file or only the plotted data can also be done from the File menu.)

Note that the statistics will be based on either CPM, CPS or µSv/h, depending on the choice applied to the graph!

The 'Hist' button will open a new window displaying the data shown in the graph as a histogram with a Poisson Distribution fit to it, plus some more statistics data, see next chapter.

Note that the histogram will either be based on CPM or on CPS, depending on the choice for the graph!

3 The Moving 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.

Count Rate Histogram with Poisson Fit

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

The value 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 fit as a quality control for your measurement.

For an introduction to Poisson Distribution and its statistics see my "Potty Training for Your Geiger Counter" article on Sourceforge https://sourceforge.net/projects/geigerlog/

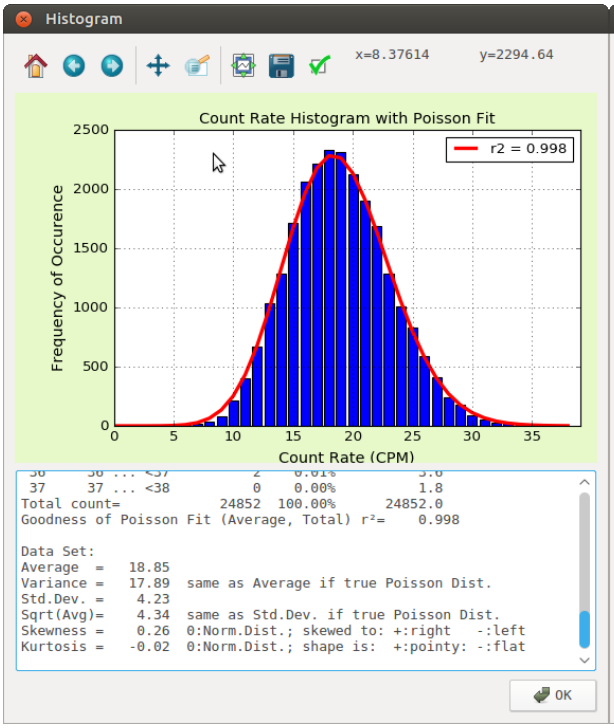


Fig. 5 Histogram of Low Count Rate

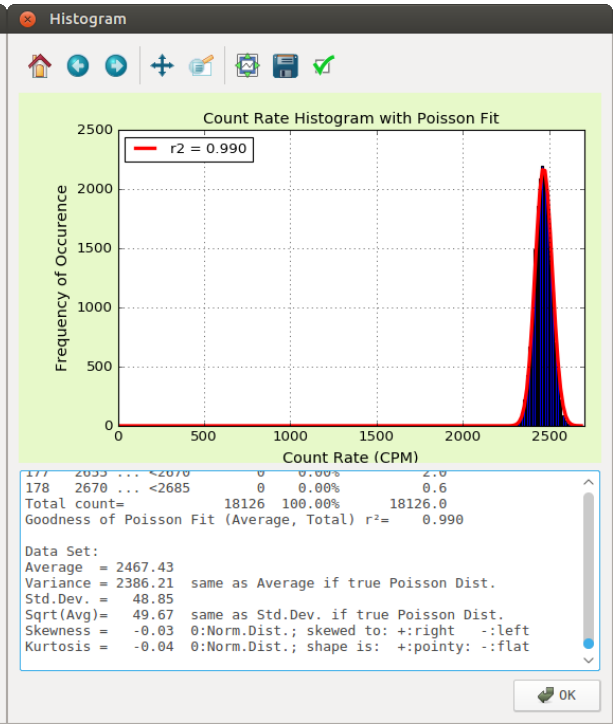


Fig. 6 Histogram of High Count Rate

FFT Analysis of Count Rate

The FFT (Fast Fourier Transform) allows to analyze a time dependent signal, like the Count Rate, for any periodic signal hidden within the data. An example is given in fig. 7 using the same data as used in fig. 6. The data were recorded by logging in the CPM mode.

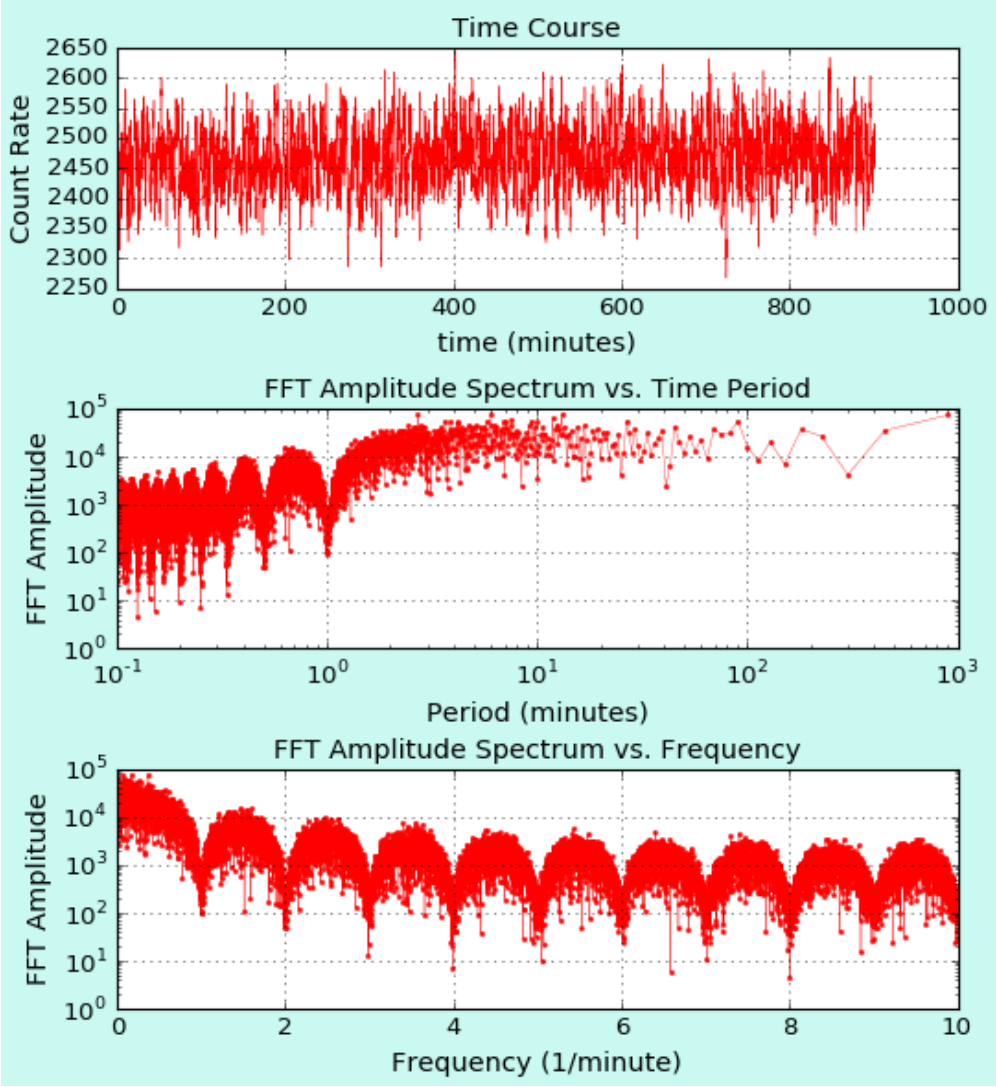


Fig. 7 FFT Analysis of High Count Rate Measurement

By looking at the Time Course of Count Rate vs. time (upper panel) I expected to find a signal with a period of 1 or more hours. However, in the range of (middle 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 panel) of 1/minute. The frequency plot (bottom panel) clearly also shows all the harmonics of this frequency.

The effect is independent of count rate (same pattern at background) and sampling time (< 30sec).

As there is a dip at 1 min, meaning there is a lack of signal “strength”, an explanation might be that the counter in CPM mode needs some time every minute to get his stuff stored away, and therefore misses a little bit of counting.

Starting GeigerLog with Options

Select menu Help → Options to print this text to the NotePad:

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

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

Options:

    -h, --help          Show this help and exit
    -d, --debug         Run with printing debug info. Default is debug = False
    -v, --verbose       Be more verbose. Default is verbose = False
    -V, --Version       Show version info and exit
    -p, --port name     Sets the USB-Port to name. Default is name = /dev/geiger
    -b, --baudrate N    Sets the baudrate to N. Default is N = 57600.
        -style name     Sets the style; see Commands showstyles
                        Default is set by your system

Commands:

    showstyles          Show a list of styles available 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

To watch debug and verbose output start the program from the command line in a
terminal. The output will print to the terminal.
```

Bugs and Problems

If you do encounter any bugs or problems please report via the project GeigerLog site at Source-forge: https://sourceforge.net/projects/geigerlog/.

Appendix A

Look & Feel

Both the scripts and the bundle use some resources which exist on your computer independently from GeigerLog. Those may differ between computers. This is mainly the “style”, but also the “fonts” available on a system. Both largely determine the Look & Feel of a software.

Style

Most styles look at least acceptable, and generally the default will be ok. But if the default style doesn’t please you, select a different one. Start GeigerLog with:

```
./geigerlog showstyles
```

to get a list of style available on your computer, which should be similar to this one:

```
Breeze, Cleanlooks, Plastique, Windows, GTK+
```

My preference is in the order listed; yours may be different. To use a style start GeigerLog like this (Note: single-dash before ‘style’):

```
./geigerlog -style Cleanlooks
```

Fonts

GeigerLog will select suitable fonts; they cannot be selected by the user. Well suited are fonts from font families “Sans Serif” and “DejaVu Sans Mono”. Install them if you want different fonts.

System Info

For more details on style, fonts and other topics start GeigerLog and select Help→ System Info.

Appendix B

Connecting Geiger Counter and Computer

BACKGROUND: Counter and computer are connected with a USB cable, but the connection is actually a classic serial connection. The translation between USB and serial is done by an USB-to-Serial chip in the electronics of the counter. And while serial connections today are much faster than in the good old days of the teletype, they are slow by today’s standards. In the GMC300 series the serial speed is in the order of 0.1MBit/s, while USB2 is nominal 480Mbit/s and USB3 even in the Gigabit/s range. With respect to logging, the speed is sufficient, but for other actions a faster speed would be welcome.

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

The serial port, the baud rate and the driver

The baud rate is set at the Geiger counter itself. To look it up at the counter go to its Main Menu → Others → Comport Baud Rate. The default is 57600 (older devices) or 115200 (newer devices). I

suggest to keep the default setting (I experienced occasional read errors with a GMC300 device, which seemed to have to do with the baud rate; and sometimes the counter chokes when things go too fast).

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

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

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

Using GeigerLog to find the Serial Port Settings

Connect the Geiger counter with the computer. Start GeigerLog and click menu **Help → USB Autodiscovery**. GeigerLog will test all available ports with all baud rates and report result into a pop-up dialog box. On a Linux system the result may be as shown in fig. 8.

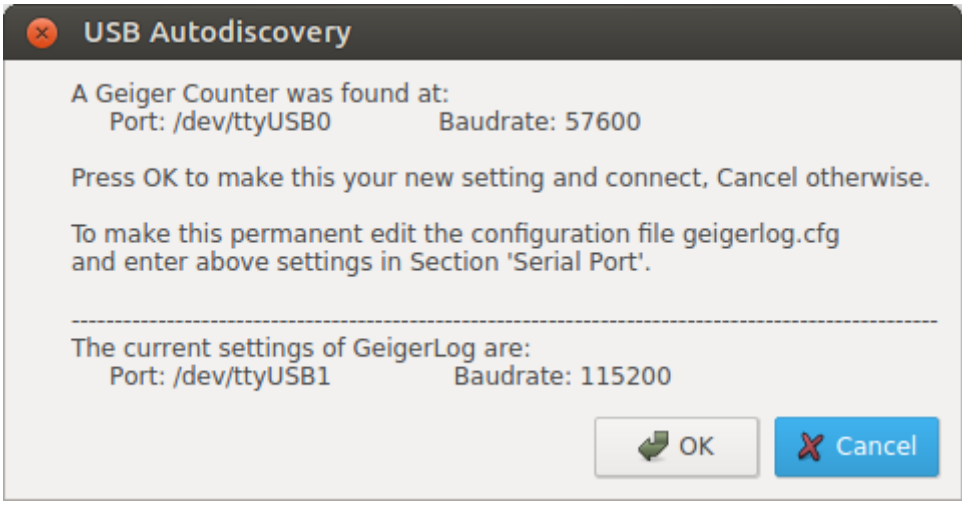


Fig. 8 USB Autodiscovery Pop-up Box

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

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

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

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

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

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

```
port      = /dev/ttyUSB0

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

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

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

and the configuration file modified to:

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

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

Appendix C

HOWTO deal with read and write permissions for the serial port when you are running the program on Linux

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

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

the output is like :

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

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

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

```
groups myname
```

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

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

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

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

1) Change permissions

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

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

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

2) Make yourself a member of group 'dialout'

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

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

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

```
groups myname
```

results in:

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

This change is permanent; also survives a reboot.

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

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

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

3) Take advantage of udev rules

In a terminal issue (as regular user):

```
lsusb
```

to get something similar to:

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

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

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

Create a file containing nothing but these two lines:

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

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

HOWEVER: The USB-ID belongs to the USB-to-serial converter chip installed in the Geiger counter. And since (to my knowledge) GQ is using the same chip an all Geiger versions, this simple rule will not allow to distinguish between them! You'll probably have to resort to option 2 above, and figure out, which /dev/ttyUSBN with N=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 Device Configuration Meanings

The device configuration 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

Furthermore, the GMC-500 series of Geiger counters seem to have a completely different configuration. No details have been disclosed gy GQ.

```
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
```

Appendix E

Internal Memory, Storage Format and Parsing Strategy

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

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

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

Date & Time Stamp

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

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

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

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

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

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

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

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

Overflow

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

The first issue to consider is that the time sequence in the memory from bottom to top is now: youngest data, followed by oldest data, which are becoming younger as you go up in memory.

Therefore GeigerLog does a final sorting of all records according to time of each record determined by the parser.

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

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

Page Boundaries

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

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

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

After the page is deleted, all bytes up to the end of the page are set to 255 (in gray). The former time fragments 19 (min) and 12 (sec) become regular counts (in white) and the remainder of the Date&Time stamp beginning at 4098 (in yellow) has now become an ASCII tag with 85 bytes of supposed ASCII code following (only 3 bytes shown) [4].

```
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) extends across a page boundary. The value at 28672 (in orange) is the Saving Mode byte, which can have values of 0, 1, 2, or 3. But it is 255.

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

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

---

4 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.

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

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. Use a program which can handle binary files, and from the *.bin file delete the segment just determined
6. The remaining *.bin file can now be opened and parsed again, and should result in a proper history. If not, repeat at step 2.

Appendix F

Firmware Differences between Geiger counter models

So far the differences seem to be relevant only for the history download.

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 no bytes are send at all by the Geiger counter.

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!

Appendix G

Calibration

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

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

GeigerLog uses the calibration of 0.0065 µSv/h / CPM.

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

I was unable to find specifications for the Geiger counter tube M4011, currently used in GQ counters. However, I established that the SBM20 tube, an old Russian Geiger tube, is similar to the M4011 at least in some aspects, and can even be used instead of the M4011 in the GMC300E+ counter, see http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4571 .

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

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

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

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

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

Looking at the gamma spectra in fig. 9 we see that Co60 is above 1 MeV, while Ra226 is mostly below 0.5 MeV. At least the SBM20 tube, according to specs, is 32% more sensitive to the lower

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

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

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

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

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

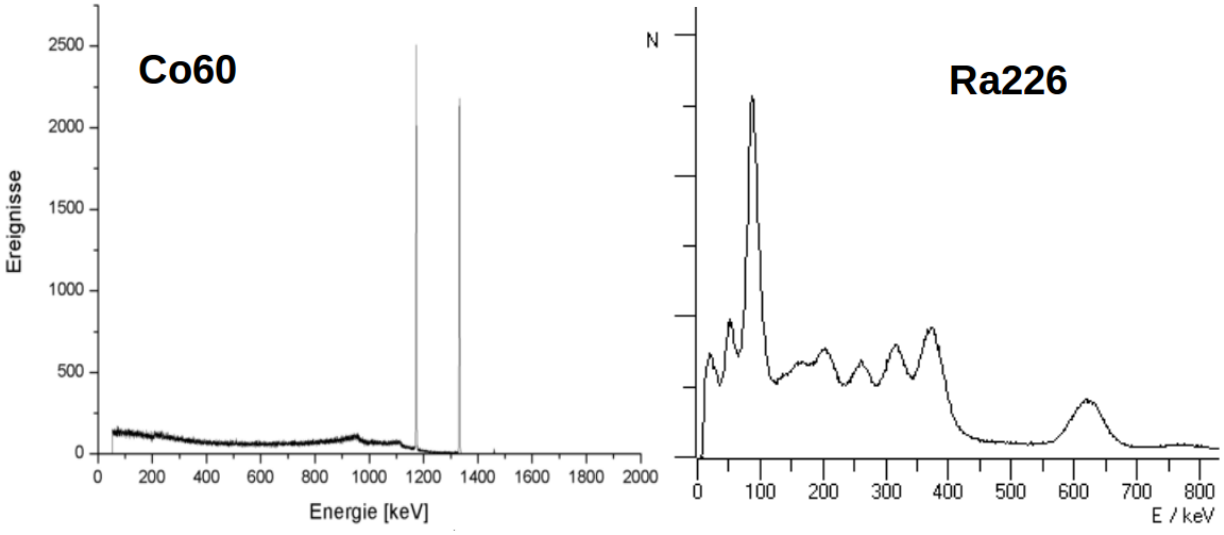


Fig. 9 Gamma Spectra of Ra226 and Co60

Appendix H

References

Louis Costrel, Accurate Determination of the Deadtime and Recovery Characteristics of Geiger-Muller Counters, 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

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

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

Appendix I

Installing the full python environment

This is tested with a **default installation** of **Ubuntu Mate 16.04.2** using only those repositories present in the default installation.

Using the scripts

When your computer is running correctly you need to install these additional packages:

- python-qt4
- python-qt4-phonon
- python-matplotlib
- python-serial
- python-scipy

You can do this with this command (all on a single line):

```
sudo apt-get install python-qt4 python-qt4-phonon python-matplotlib python-serial python-scipy
```

Using the bundle

There should be nothing extra that needs to be installed.

If it still fails?

If it still fails, 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 what modules GeigerLog misses and install those.