GeigerLog Manual

by ullix

Version 0.9.07

May 2018

What's New in GeigerLog 0.9.07 ?

- GeigerLog now uses Python 3
- Now supporting GMC-300, GMC-300E+, GMC-320+, GMC-320+V5, GMC-500(+), GMC-600(+)
- Auto-detecting Geiger Counter model, and adjusting internal settings, calculations, calibration, as well as work-arounds for firmware-bugs automatically
- Customization possible to accommodate older counters, and potentially even for new ones not yet on the market
- Supports World Radiation Maps even for devices without WiFi
- Customizable graphics (colors, line widths, line types, markers)
- Count Rate Display area click sensitive to allow manually triggered count rate measurements

Recommended Reading on the subject from the same author:

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

GeigerLog - Potty Training for Your Geiger Counter

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

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

GeigerLog - Going Banana

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

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

Table of Contents

Overview

GeigerLog is a program to use GQ Electronic's [1]) GMC-3xx, GMC-5xx, and GMC-6xx line of Geiger counters. It is based on Python, hence it runs on Linux, Windows, Macs, and others.

GeigerLog auto-detects the type of connected Geiger counter and adjusts itself to match features and/or deficiencies of the connected counter.

The two main functions of the Geiger counters are **Logging** and **History.** During Logging real time data read from the counter are printed, displayed, plotted graphically, and logged into a 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 treated just like the logging files.

All data can be shown as a plot of 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 modes. 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 only within that range and to limit the analysis of statistical properties to only those data. These ranges can be entered manually or by left/right mouse clicks.

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

Several Quality Control tests can be applied to the data. Beyond the standard statistical properties, which can be printed to the screen, a Poisson test can be applied to see how well the data fir to a Poisson distributions. Also, a FFT frequency and Autocorrelation analysis by Fast Fourier Transform (FFT) can be done to check for any cyclic effects.

Geiger counter functions (Speaker, Alarm, Saving mode, Date&Time), can be controlled and set from within GeigerLog.

The USB port used and its baud rate for the connection with the counter can be auto-discovered, and used directly to make a connection. All communication with the device is checked for errors, which do occur occasionally. The program attempts to auto-recover from an error, and continues if successful, which it is in most cases.

Several genuine recordings of Geiger counter data are included, among them a recording from an international flight, indicating count rate dependence of flight altitude and latitudinal position.

Availability

The most recent version of GeigerLog, including this manual, can be found at project GeigerLog at SourceForge: https://SourceForge.net/projects/geigerlog/. There you also find additional Geiger counter recordings and synthetic data to experiment with.

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

Introduction to GeigerLog

Installing and Starting GeigerLog

Beginning with this current release, GeigerLog version **0.9.07,** a **Python 3** environment. Future releases will all be based only on Python 3.

It was developed with Python version 3.5.2. It may run with earlier versions of Python 3.x, but this has not been tested.

In addition to Python 3 a few more modules are needed, which generally are not available in a default installation. Step-by-Step installation instructions for Python on **Linux** and **Windows 7, 8 and 10** are provided in Appendix H – Installation on page 45.

The software 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 runtime environment. This latter All-in-One package has no further installation requirements, however, currently **only a Linux bundle** exists!

The scripts

The scripts are only a few hundred kB big and require a proper Python environment (see above). If this condition is met, GeigerLog will run on Linux, Windows, Mac, and other systems.

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

The bundle

NOTE: the bundle is no longer being offered due to lack of interest. If you do need it, please, contact me by email via SourceForge.

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.

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

Installing

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).

Starting the scripts

1. Make sure you have Python installed, including all modules called in the scripts.
2. Start GeigerLog with

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

Starting the bundle

1. 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 – Look & Feel on page 30.

The GeigerLog Window

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

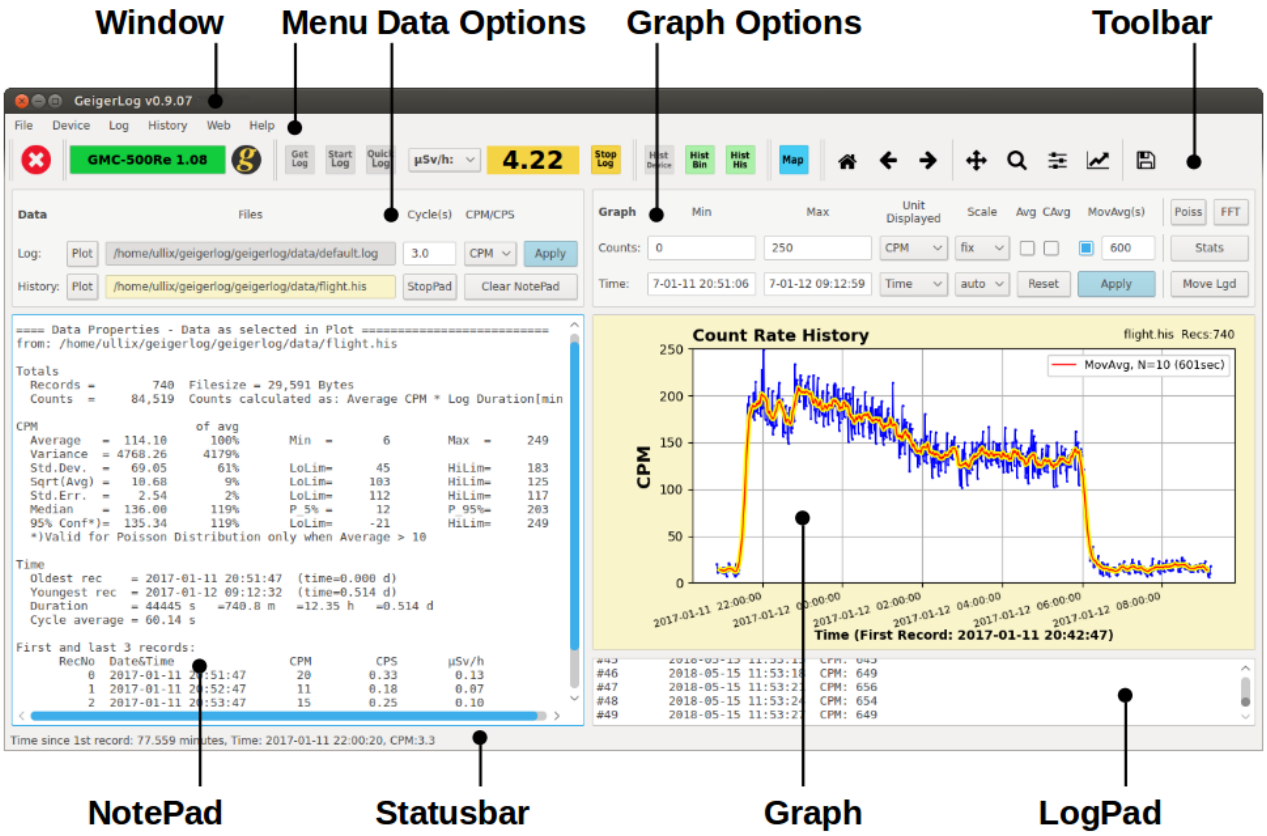


Fig. 1 GeigerLog Window with Annotations

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

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 Geiger counter device in order to analyze **existing** data:

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'.

In the graph, do a mouse-**left**-click somewhere on the vertical line near 8 h, and a mouse-**right**-click on the vertical line near 10 h. Note that the Time Min and Max fields in the Graph 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 'Stats' button in the Graph Options area. 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 'hour' under Time Unit, then **left**-click on the vertical line near 10 h, and Apply. Then click button '**Poiss**', 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 right-most icon 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!

Connecting GeigerLog to Your Geiger Counter

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**. Alternatively, and easier, click the long gray button with the label "Click for Connection".

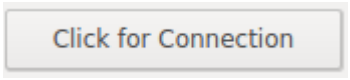
- If successful, a printout into the NotePad **in black** will confirm the connection. You are set to continue.
- If unsuccessful, a printout **in red** will tell you the reason. You will be advised to run **USB Autodiscovery** from menu **Help**. If you are lucky, a dialog box will pop-up and tell you 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.

> **WARNING:** If you have other devices – besides any Geiger counter – connected with a **USB-to-Serial** adapter (which is what your counter has built-in) it is possible that the other devices are significantly disturbed by the USB Autodiscovery!
>
> It is recommended to halt or switch off those other devices BUT LEAVE THEIR USB CABLE CONNECTED, and only then start the USB Autodiscovery. Any other native USB devices (mouse, keyboard, printer, ...) won't be impacted by the test.

  For more complicated situations see Appendix B – Connecting Geiger Counter and Computer on page 30.

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

disconnected 

to connected 

, or 

, or else, with the label text telling you, what device it had found and which firmware revision the counter has. This is how the counter identifies itself.

Note that you can easily toggle the connection state by clicking this button in the toolbar!

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 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! When the icon is gray then GeigerLog has not yet been able to determine the power state of the counter.

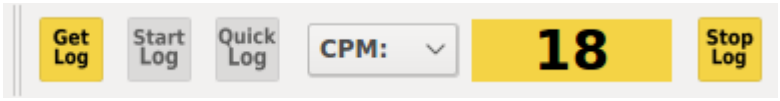
Your Geiger Counter Model

GeigerLog works the same for all 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 had clicked the connection button this same button shows the correct Geiger counter and firmware.

If this is not the case, then you have to customize your model by modifying the configuration file `geigerlog.cfg` in its **Device** section. This is explained in Appendix F – Firmware Differences on page 42.

Logging

Any logging is controlled by GeigerLog, not by the counter. 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, µSv/h, mSv/d, or mSv/a:



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 file default.log 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 the StartLog button instead.

The Count Rate Display area (which in the above picture the yellow field showing '18') can be used to manually trigger an instantaneous count rate measurement by a mouse left-click. It responds by showing in the NotePad area the result, like:

```
==== Manually Triggered Log Value ================================
CPM :    653   =   4.24 µSv/h   =   0.10 mSv/d   =   37.18 mSv/a
```

This feature is helpful, if you have set a long cycle time and want to see an immediate value, or if want to quickly glance at what this means in other does rate units.

A note on the logging cycle

The Geiger counter needs less than 1 ms (millisecond) [2]) 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.

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 now clickable Count Rate Display (see above) makes it easier to use long cycle times, and get a measurement whenever you need one.

This **oversampling** – sampling more often than really needed – has consequences for certain properties of the data, see Quality Control - FFT Analysis of Count Rate on page 27.

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.

The logging is controlled by the commands in Menu – Log.

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.

---

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

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 – Internal Memory, Storage Format and Parsing Strategy).

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 ring-buffer (see Appendix E – Internal Memory, Storage Format and Parsing Strategy) storage principle becomes effective, and the memory is overwritten beginning at the bottom. In this situation the whole memory is filled with data, and there will never be 2 pages of empty values. Hence the whole memory will be read! If you don't need the content of the memory, I suggest to erase it every once in a while.

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`!

Handling the History is controlled by three buttons in the toolbar, and the commands available in Menu – History. Hist Device downloads the history from the counter to your computer, creates a plottable file, and shows the plot. Buttons 'Hist Bin' and 'Hist His' start files created by a previous history download, see more at Menu – History.

Starting GeigerLog with Options

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

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

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

Options:
    -h, --help          Show this help and exit
    -d, --debug         Run with printing debug info
                        Default is debug = False
    -v, --verbose       Be more verbose
                        Default is verbose = False
    -V, --Version       Show version status and exit
    -p, --port name     Sets the USB-Port to name
```

```
                        Default is name = /dev/ttyUSB0
    -b, --baudrate N    Sets the baudrate to N
                        Default is N = 57600
    -R  --Redirect      Redirect stdout and stderr to
                        file geigerlog.stdlog (for debugging)
        -style name     Sets the style; see Commands:
                        'showstyles'
                        Default is set by your system
Commands:
    showstyles          Show a list of styles avail-
                        able on your system and exit.
                        For usage details see manual
    keepFF              Keeps all hexadecimal FF
                        (Decimal 255) values as a
                        real value and not an 'Empty'
                        one. See manual in chapter
                        on parsing strategy
    devel               Development settings; careful!
                        see program code

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

With the Redirect option all output - including Python error
messages - will go into the redirect file.
```

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, which are essential for debugging. With the redirect option another log file `geiger-log.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.

The Graphical User Interface

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 Poisson Test — In a pop-up window show a histogram 'Count Rate Histogram with Poisson Fit' for data of the whole file
- Show Plot Poisson Test — 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 & Autocorrelation — In a pop-up window show a FFT and autocorrelation analysis of the Count Rate data of the whole file
- Show Plot FFT & Autocorrelation — In a pop-up window show a a FFT and autocorrelation analysis of the Count Rate data of those data currently shown in the plot
- Exit — Exit the program (will be prevented if Logging is ongoing; stop Logging first)

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 – The Device Configuration Meanings 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 – The Device Configuration Meanings.
- Switch Power ON — Switches the device power ON (as if pressing the Power button on the device).
- Switch Power OFF — Switches the device power OFF (as if pressing the Power button on the device).
- Switch Alarm ON — Switches the device alarm ON.
- Switch Alarm OFF — Switches the device alarm OFF.
- Switch Speaker ON — Switches the device speaker ON.
- Switch Speaker OFF — Switches the device speaker OFF.
- 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 device 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, it 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 – Web

- Update Radiation World Maps — Upload your current data to the Radiation World Maps, see Radiation World Maps on page 23.

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
- 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 – Connecting Geiger Counter and Computer for more details.

- 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

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.

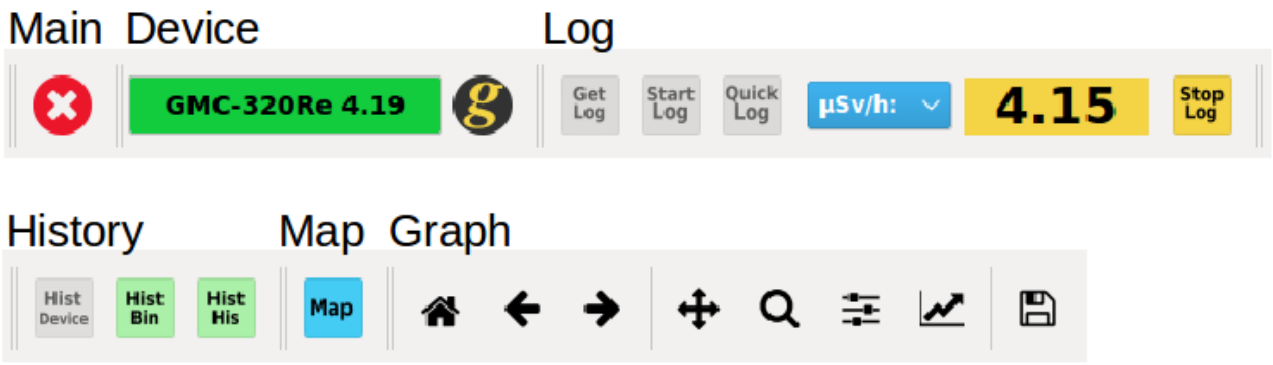


Fig. 2 The toolbars

- Main — - Exit GeigerLog
- Device — - Toggle Connection to Geiger Counter, indicating connected counter
  - 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; also: a mouse left-click in this field triggers an immediate count rate measurement to be shown in the NotePad in multiple count rates units
  - Stop Logging
- History — - Get History from device (possible only without Logging; counter gets confused with parallel activities)

- Get History from binary file
- Get History from parsed file

- Map — - Upload your current data to the Radiation World Maps,
  see Radiation World Maps on page 23.
  You must be Logging for the Map icon to become active (=blue)
- 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

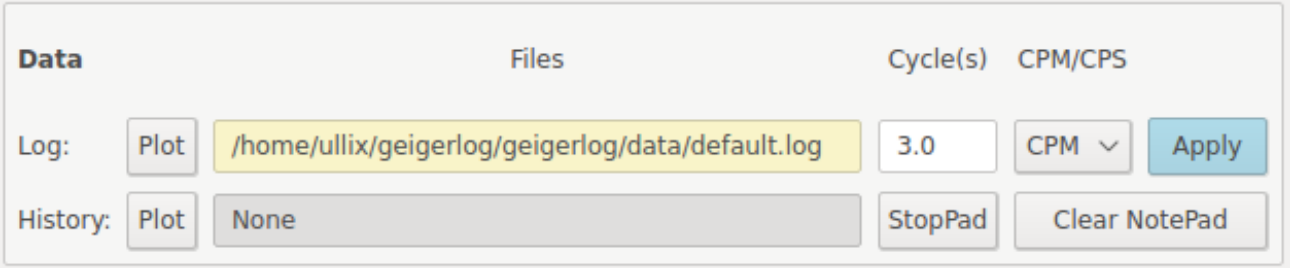


Fig. 3 Data Options

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)** entry field 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 **StopPad** button allows you to stop a printout currently going into the NotePad area, e.g. if you clicked the menu option Log → Show Log Data, which may run a very long time for very big log files.

The **Clear NotePad** button clears the NotePad area below.

Graph Options

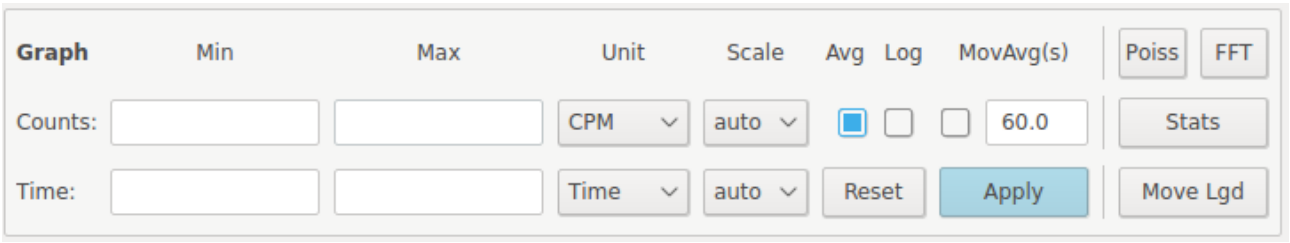


Fig. 4 Graph Options

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 the Geiger counter. E.g., for a GMC-300 series this is CPM * 0.0065 = µSv/h [3]). This setting can be changed in **Device** section 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 and the Standard Error will be shown in the graph's legend.

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/geiger-log/.

If the checkbox 'Log' is checked, then the logarithm to base 10 of the count data will be plotted. Helpful if your log file contains very low and very high counts. Note that this can also be achieved by clicking the Graph Toolbar icon  and then selecting Scale Log for the Y-axis (you can also do further modifications via this toolbar icon).

A moving average [4]) (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 us-

3 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 43.

4 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.

ing 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 ‘Stats’ , ‘Poiss’, and ‘FFT’ buttons are part of the tools for the Quality Control of your data, see chapter Quality Control of your Data on page 25.

- The ‘Stats’ button will print statistics on the plotted data into the NotePad area. Note that the statistics will be based on either CPM, CPS or µSv/h, depending on the choice applied to the graph!
- The ‘Poiss’ 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. Note that the histogram will either be based on CPM or on CPS, depending on the choice for the graph!
- The ‘FFT’ button will open a new window displaying the data shown in the graph as a FFT (Fast Fourier Transform) and Autocorrelation analysis.

Radiation World Maps

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

Among the latter ones are:

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

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

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

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

When you are logging, the toolbar icon 'Map' turns blue 

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

You will see a confirmation printed to GeigerLog's NotePad area, including the response of the website.

A word of caution

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

> **CPM:** The property depicted on the map is CPM, which is the worst possible base on which to compare different counters, which may have different tubes and even different tube numbers, and therefore totally different calibration factors to translate from CPM to a true dose rate like measured in µSv/h. This is like a worldwide reporting of temperatures as either Fahrenheit, or Celsius, or Reaumur but not telling which is which. The only meaningful basis for comparisons is the dose rate based on units of Sievert per time intervall (µSv/h, or nSv/h).
>
> **Quality Control:** As far as I can see there is no quality control of the data! Nothing prevents users from putting a strong radioactive source in front of their detector, and pushing these data to the web. In fact, you don't even need a counter, and don't even need GeigerLog, but can enter any data you wish manually! I don’t want to mess with GQ’s map, so I haven’t tried to enter

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

Poor data will quickly destroy any value of those sites.

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

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

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

Governmental sites like this Swiss site provide even only DAILY averages of quality controlled data!

Occupational Radiation Limits

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

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

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

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

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

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

Quality Control of your Data

The ‘Stats’ , ‘Poiss’, and ‘FFT’ buttons in the Graph Options area help you to check the quality of your data.

Statistics

Clicking the ‘Stats’ button will print some standard statistics into the NotePad area, which will look like this printout from a high count rate experiment:

```
==== Data Properties - Data as selected in Plot ==========================
from: /home/ullix/geigerlog/data/Gas_mantle-M4011-2017.log

Totals
  Records =     27,664  Filesize = 940,982 Bytes
  Counts  =  3,830,028  Counts calculated as: Average CPM * Log Duration[min]

CPM                      of avg
  Average   = 2769.06      100%       Min  =    2586        Max  =    2973
  Variance  = 2625.51       95%
  Std.Dev.  =   51.24        2%       LoLim=    2718        HiLim=    2820
  Sqrt(Avg) =   52.62        2%       LoLim=    2716        HiLim=    2822
  Std.Err.  =    0.31        0%       LoLim=    2769        HiLim=    2769
  Median    = 2768.00      100%       P_5% =    2685        P_95%=    2854
  95% Conf*)=  100.43        4%       LoLim=    2669        HiLim=    2869
  *)Valid for Poisson Distribution only when Average > 10

Time
  Oldest rec    = 2017-06-24 11:46:45  (time=0.000 d)
  Youngest rec  = 2017-06-25 10:49:54  (time=0.961 d)
  Duration      = 82989 s   =1383.2 m   =23.05 h   =0.961 d
  Cycle average = 3.00 s

First and last 3 records:
      RecNo  Date&Time                CPM         CPS       µSv/h
          0  2017-06-24 11:46:45     2771       46.18       18.01
          1  2017-06-24 11:46:48     2762       46.03       17.95
          2  2017-06-24 11:46:51     2780       46.33       18.07
      27661  2017-06-25 10:49:48     2807       46.78       18.25
      27662  2017-06-25 10:49:51     2782       46.37       18.08
      27663  2017-06-25 10:49:54     2784       46.40       18.10
```

Note that the statistics will be based on either CPM, CPS or µSv/h, depending on the choice applied to the graph; here the default CPM had been chosen.

As a first easy check look at Average and Variance – they should be about the same (is the case here), unless you had intentionally modified measuring conditions during a recording.

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

This ‘Stats’ button will only print the statistics of the data currently shown in the plot. Obviously, if all the data are shown, then the statistics does cover all the data. However, you can always go the

File menu and select ‘Print File Stats’ to print all data in the whole file irrespective of the Graph settings!

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 latter uses the same data as used in the Statistics printout.

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 another quality control tool for your measurement.

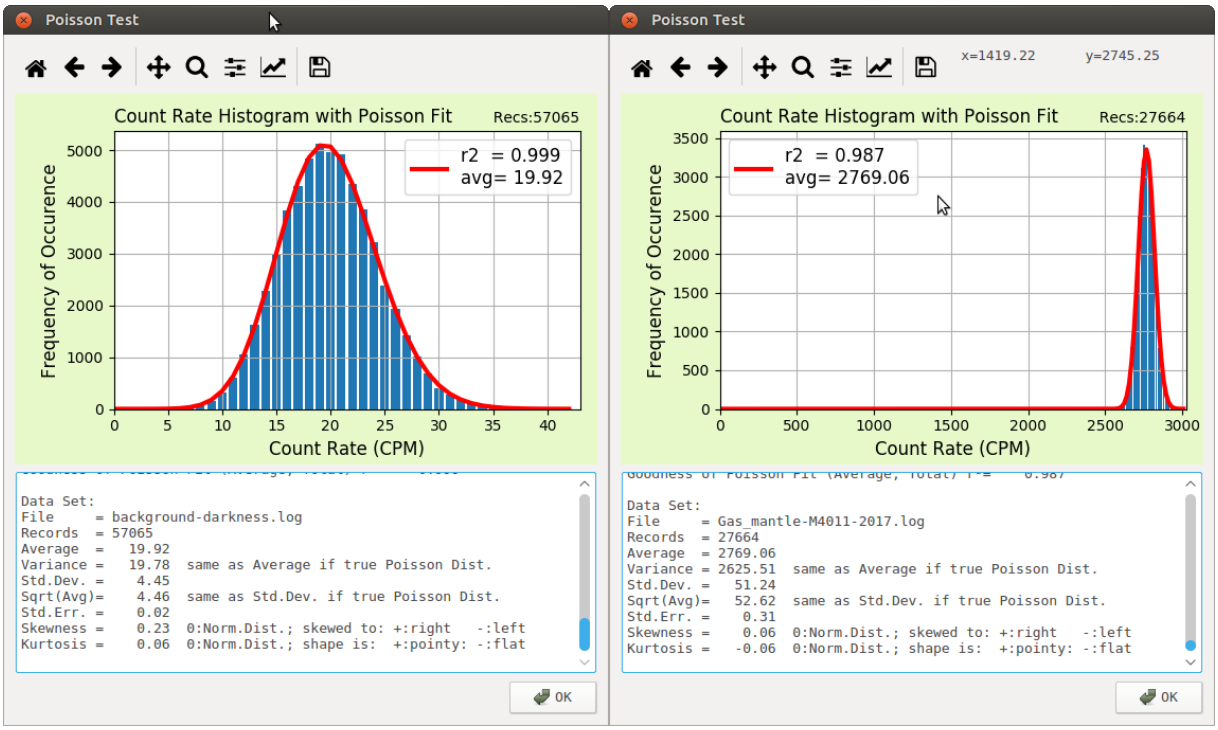


Fig. 5 Histogram of Low Count Rate *Fig. 6 Histogram of High Count Rate*

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

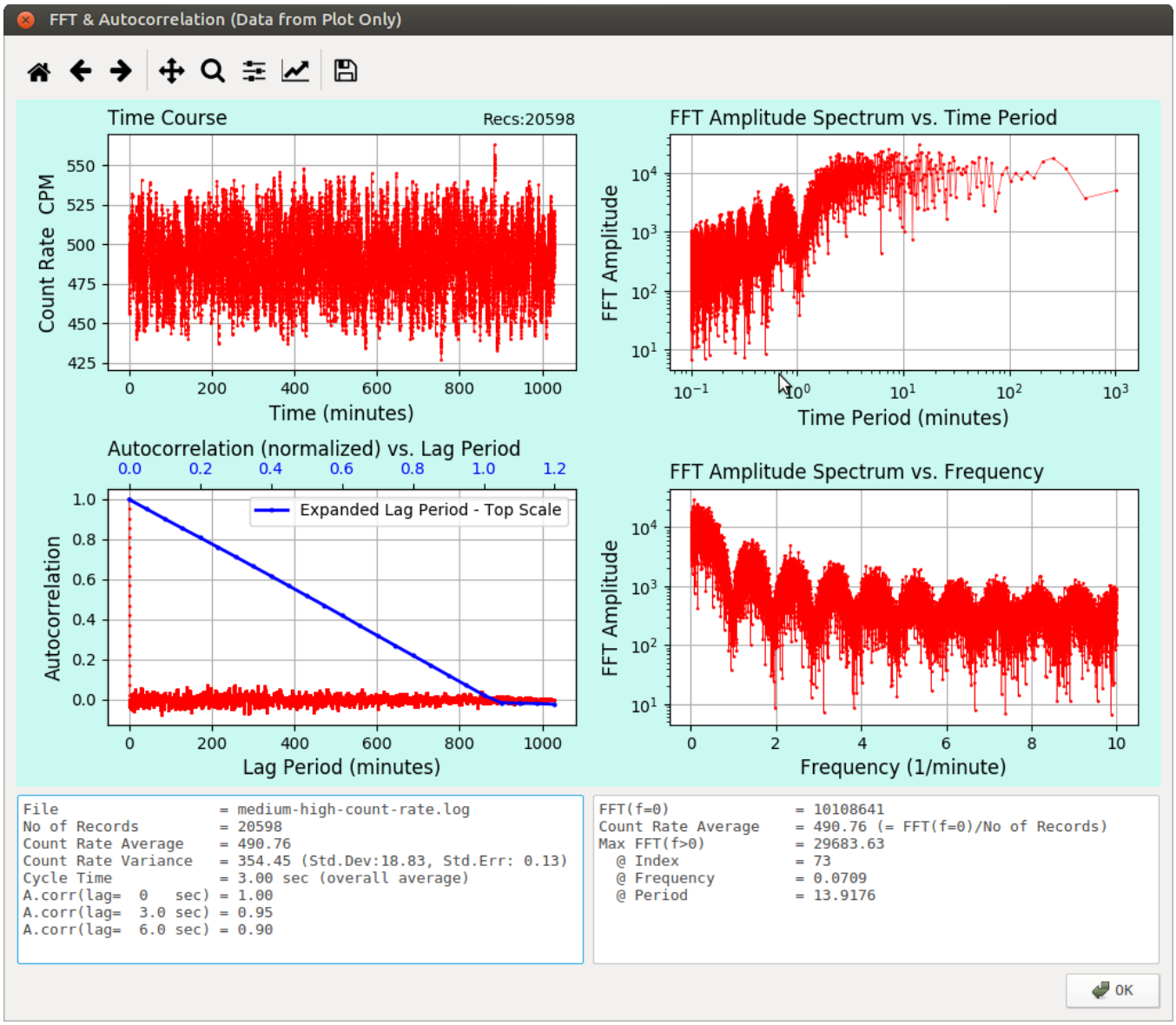


Fig. 7 FFT Analysis of Medium Count Rate Measurement

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

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

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

This can be nicely demonstrated using synthetic recording. In fig. 8 the upper panels show the signals in the time domain from Poisson White noise at average CPS=2.5 (upper left), a rectangle of 1 min at value 1 and value 0 for the remaining 19988 counts (upper middle), and the convolution of these two signals (upper right), resulting in average CPM=150. The bottom panels show the corresponding FFT spectra, white noise, a 1/min frequency and harmonics, and the mix of the two.

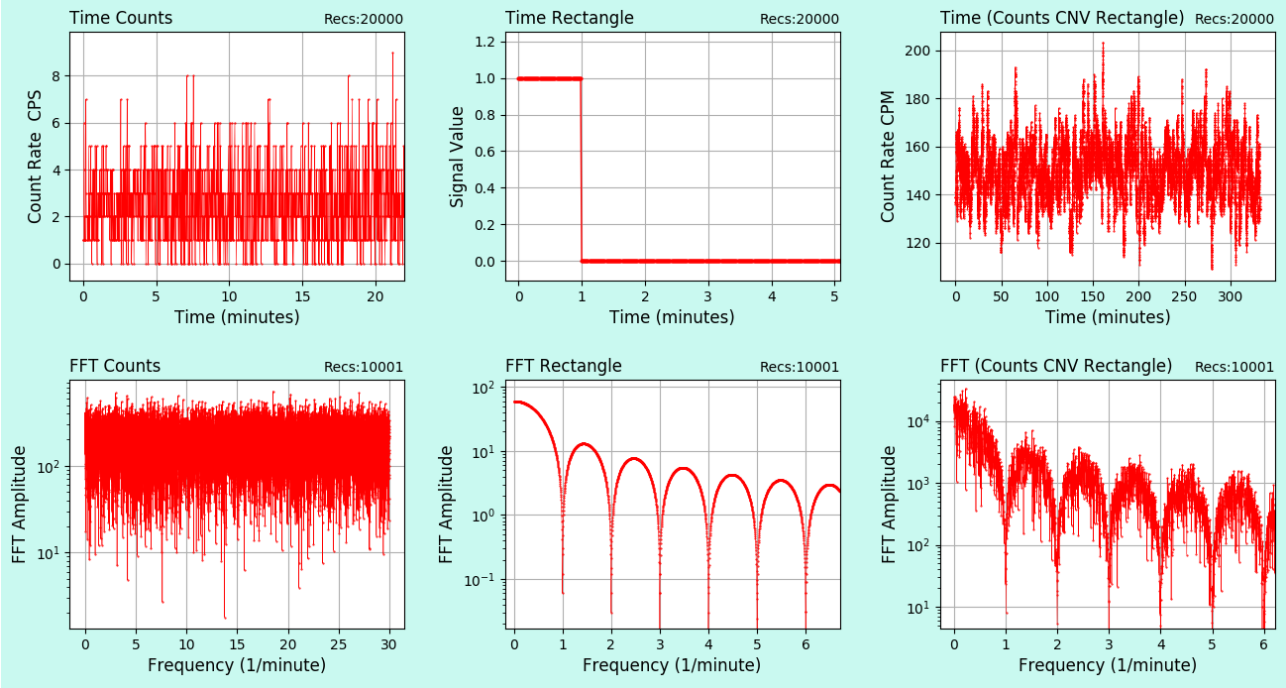


Fig. 8 Demonstration of the impact of convolution on an FFT spectrum (Synthetic data, CPS=2.5)

Oversampling does no harm; but it must be accounted for when autocorrelation plays a role.

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 45 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 13.

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

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 with the command ‘`devel`’ 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 GMC-300 series the serial speed is in the order of 0.1 MBit/s, while USB2 is nominal 480 Mbit/s and USB3 even in the Gigabit/s range. With respect to logging, the speed is sufficient, but for other actions a faster speed would be welcome.

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

The serial port, the baud rate and the driver

The baud rate is set at the 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 GMC-300 device, which seemed to have to do with the baud rate; and sometimes the counter chokes when things go too fast).

On Linux the driver is already part of the system. On Windows and Mac a driver must be installed. Drivers may be available for download on the GQ website [6]).

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

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

Using GeigerLog to find the Serial Port Settings

Connect the Geiger counter with the computer. Start GeigerLog and click menu **Help → 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. 9.

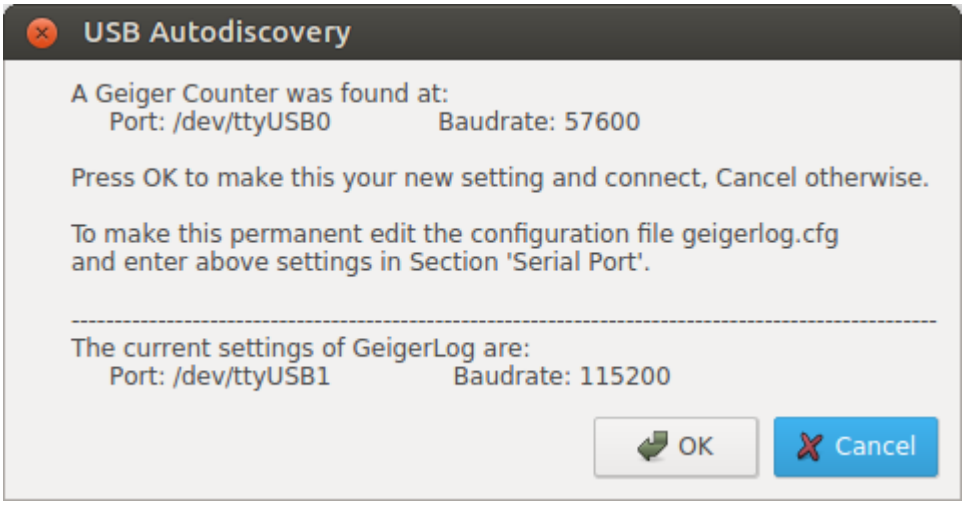


Fig. 9 USB Autodiscovery Pop-up Box

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

It is recommended to halt or switch off those other devices BUT LEAVE THEIR USB CABLE

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

CONNECTED, and only then start the USB Autodiscovery. Any other native USB devices (mouse, keyboard, printer, ...) won't be impacted by the test.

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

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

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

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

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

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

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

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

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

and the configuration file modified to:

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

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

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

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

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

the output is like :

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

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

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

```
groups myname
```

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

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

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

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

1) Change permissions

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

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

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

2) Make yourself a member of group 'dialout'

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

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

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

```
groups myname
```

results in:

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

This change is permanent; also survives a reboot.

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

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

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

3) Take advantage of udev rules

In a terminal issue (as regular user):

```
lsusb
```

to get something similar to:

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

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

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

Create a file containing nothing but these two lines:

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

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

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

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

Appendix D – The 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 29).

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 – 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) [7].

```
 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
```

7 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.

```
 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. If you see negative counts – this is the reason.

The 255 value

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

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

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

```
./geigerlog keepFF
```

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

Correcting a Wrong History

It is an annoying procedure. The following is suggested:

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

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

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

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

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

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

- Both Logging and History download is working on all models
- Reading the calibration works on all series
- Reading and Setting Geiger counter configurations like, alarm, speaker, power status, History saving mode works for all series. However, they do not work reliably, not even for the old 300 series counters: every now and then a function fails, which always turned out to be due to an unexpected timeout of the counter. This is an issue of the counter's firmware! GeigerLog attempts to correct the failure, and is mostly, but not always successful. Look for the output printed into 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 area of 256 bytes. There is confusion around the meaning of each entry (see Appendix D – The Device Configuration Meanings, page 34), 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).

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 [8]):

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

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

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

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

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

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

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

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

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

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

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

Looking at the gamma spectra in fig. 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.

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

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!

Appendix H – Installation

General

This version of GeigerLog requires Python version 3.x. While it has been developed on Python 3.5.2, it may work on earlier versions, though this has not been tested. The latest Python version is currently 3.6.5.

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' [9]) also installed! More on Pip is here: https://pip.pypa.io/en/stable/ .

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 distribution and your history of installation, one of the two can probably be started with 'python', while the other needs to be started with 'python2' or 'pathon3'. Likewise with pip. To find out your situation, look at the output of:

```
python -V
python2 -V
python3 -V
pip -V
pi2 -V
pip3 -V
```

We will now assume that Python 3 and Pip3 is installed, and you must use the commands '**python3**' and '**pip3**' to start your Python 3 and matching Pip.

Modules

Now you have a working Python Installation. In addition to it, you also need these Python modules:

- qt4
- qt4-phonon
- matplotlib
- numpy (at least version 1.14!)
- serial
- scipy

9 Pip is a recursive acronym that can stand for either "Pip Installs Packages" or "Pip Installs Python".

While for all modules a ‘recent’ version should suffice, the module numpy MUST BE at least of version 1.14!

The qt4* modules will probably have to be installed via the distribution tools (like apt-get for Linux), but for the other modules you may be better off using pip.

Using pip

It is often easier – and mostly gives you a more recent version – to use the program ‘pip’ instead of the standard tool of your distribution (like apt-get for Ubuntu). E.g., to install numpy, do (you need root privileges for any installation with pip) :

```
pip3 install numpy
```

If numpy is already installed, but the version is not at least 1.14, then upgrade numpy with pip by

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

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

To just show the version of a module, use `show`:

```
pip3 show numpy
```

If GeigerLog fails to run?

Modules may still be missing, or are deprecated. 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. Sometimes the distribution installs deprecated packages; update them with pip.

If pip complains that it can’t do an update, you may have to remove this package first with the distribution tools, like for Ubuntu:

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

before reinstalling with pip. (Such was the case for the serial module in the Mint distribution.)

Installation of an Editor

GeigerLog has been developed on the editor Geany, which I do recommend. If you don’t have a good editor yet, consider installing Geany. Get `geany-1.31_setup.exe` or later from: https://www.geany.org/Download/Releases

Linux - Installation

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

Get the additional modules

When your computer is running correctly with the proper version of Python, you need to install the additional modules listed above. You can do this with this command in one step (all on a single line):

```
sudo apt-get install python3-qt4 python3-qt4-phonon python3-matplotlib python3-serial python3-scipy
```

Using pip

Install pip for Py3:

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

To check the numpy version:

```
pip3 show numpy
```

To upgrade numpy with pip:

`sudo -H pip3 install numpy --upgrade` (NOTE: **2** dashes before upgrade!)

Installation of GeigerLog

Copy the `geigerlog-scripts-xyz.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 -d
```

or create a starter to be put in the panel.

(The '-d' option lets GeigerLog start in Debug-Mode; you'll find the output in the terminal and in the program log file `geigerlog.proglog`)

Windows - Installation

The Python 2 version of GeigerLog was verified to work with Microsoft-provided Virtual Machines for Virtualbox of **Windows 7**, **Windows 8.1**, and **Windows 10**. GeigerLog worked in all of them. It is expected the Py3 version works just as well.

Verify the current Python installation status on your Windows machine

On a fresh Windows installation there is probably no Python installed. The steps in the "Verify ..." chapter above will have provided you with the installation status.

If you do have Python installed, it is recommended that you upgrade if your version is older than the one suggested below. But at any rate, you should have `pip` installed! If you have not, and you find no other way to add `pip`, then de-install your current Python version – even if it is a newer one – and do a new install of the version below.

Installation of Python 3.6.5

Download from here: https://www.python.org/downloads/release/python-365/

Attention:

in the installer make sure to set a checkmark to the option **„Add python.exe to path“** !

The batchfile to start GeigerLog '`geigerlog.bat`' also uses '**`python3`**'; change it if necessary.

Installation of Modules

If your CMD window is still open then close it now and re-open (Really!). Type:

```
python3 -V                  → expected: =3.6.5
pip3 list                   → expected: pip=10.0.1,
                                        setuptools=39.1.0
pip3 install pip --upgrade  → expected: already up-to-date
pip3 install pyserial       → expected: =3.4
pip3 install matplotlib     → multiple installs, expected a.o.:
                                         matplotlib = 2.2.2
                                         numpy = 1.14.3
pip3 install scipy          → expected: =1.1.0
pip3 list                   → expected: the above results
```

Any modules already installed should be upgraded:

```
pip3 install <modulename> --upgrade
```

Installation of PyQt4

Unfortunately the most recent versions of PyQt4 no longer come with a Windows installer, and a compilation is quite difficult. Fortunately, these slightly older versions are still available and sufficient:

Download from here: https://SourceForge.net/projects/pyqt/files/PyQt4/PyQt-4.11.4/
32bit: PyQt4-4.11.4-gpl-Py2.7-Qt4.8.7-x32.exe
64 bit: PyQt4-4.11.4-gpl-Py2.7-Qt4.8.7-x64.exe

In the installer: make sure that **every option** is checked!

Installation of GeigerLog

It is suggested to place GeigerLog directly under `c:\`. Then:

- copy the `geigerlog-scripts-xyz.zip` file to c:\
- unpack here

The unpacking will have created the folder `c:\geigerlog` with the required content.

Start GeigerLog from CMD window with:

```
python3 c:\geigerlog\geigerlog -d
```

or, more conveniently, create a shortcut to the file `geigerlog.bat` (in your geigerlog folder) and place the shortcut on your desktop. The file has the sole content:

```
python3 c:\geigerlog\geigerlog -d
```

Adapt it to your situation if you installed GeigerLog elsewhere.

(The '-d' option lets GeigerLog start in Debug-Mode; you'll find the output in the command window and in the data file `geigerlog.proglog`)

Mac – Installation

The following info 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 topics from installation to starting from a terminal and for Finder.

A version of Py2 is available pre-installed since Mac OSX10.8, but not Py3, apparently not even in the newest Macs. To install Py3 see instructions under the above link. A "universal binary" build of Python, which runs natively on the Mac's new Intel and legacy PPC CPU's, is there available.

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

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

Now that you have Python working, verify your installation status on your machine with the commands given above under topic General of this chapter.

Next is the installation of PyQt4, offered here: https://sourceforge.net/projects/pyqt/files/PyQt4/. Download the latest version. Each requires an installation as explained in its README file.

Using pip

Pip should be available after the installation of Py3. Install remaining modules with pip:

```
pip3 install matplotlib        → expected: =2.2.2
pip3 install numpy             → expected: =1.14.3
pip3 install serial            → expected: =3.4
pip3 install scipy             → expected: =1.1.0
```

To check the numpy version:

```
pip3 show numpy
```

To upgrade numpy with pip:

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

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

Installation of GeigerLog

Copy the `geigerlog-scripts-xyz.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 -d
```

(The '-d' option lets GeigerLog start in Debug-Mode; you'll find the output in the terminal and in the program log file `geigerlog.proglog`)

Appendix I – License

GeigerLog is licensed under GPL3.

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