GeigerLog Manual

by ullix

Version 0.9.08

July 2018

What's New in GeigerLog 0.9.08 ?

- Support for RadMon+ Series devices, which offer Geiger counter function combined with a sensor for the environmental variables Temperature, Air-Pressure, and Humidity
- Implementing of IoT (Internet of Things) support
- Updates on GMC-Device support, now including 4-byte CPM and CPS values, support of double tube devices (GMC-500+) for both CPM and CPS
- GMC-Device history support for 3 and 4 byte storage values
- GUI overhaul with more flexible graphing for up to 10 variables on 2 Y-Axis
- Slider-adjustable sub-windows for Graph, NotePad and LogPad
- Input values can be transformed with a powerful interpreter handling algebraic formulas with mathematical functions (log, trig, sqrt, power,…) to correct offsets, linearize values, convert air-pressure to sea level, account for dead time losses of Geiger counts, and more
- Overall improved handling of graph, statistics, analysis

Recommended Reading on the subject from the same author:

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

GeigerLog - Potty Training for Your Geiger Counter

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

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

GeigerLog - Going Banana

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

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

Table of Contents

Overview

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

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

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

Currently Supported Devices:

GMC Devices:

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

RadMon+ Devices:

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

Main Operations:

The main operations of GeigerLog are **Logging** and **Displaying**, whereby the data are:

1. read from the connected devices
2. saved into a file
3. printed as a numeric list to the screen
4. and displayed as a live graph, auto-updated during logging

All data are shown as Time Course, i.e. as a plot of value versus time. The scales are set automatically, but can be changed manually.

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

Time ranges can be set to plot data only within that range and to limit the quality control analysis to only those data. These ranges can be entered manually or by left/right mouse clicks. The time can be shown as Time-of-Day, or time since first record in units of sec, min, hours, days, or auto-selected in auto mode.

Geiger counter data are shown in units of CPM / CPS or µSv/h, all on the left Y-axis.

Environmental variables are shown on the right Y-axis; for temperature the choice is between °C and °F.

All manipulations of the plots, and all data analysis can be done during ongoing logging without disturbing it. The graphs can be stretched, shifted, and zoomed for details, and saved as pictures in various formats (png, jpg, tif, svg, ...).

1 GQ Electronics LLC, 5608 Delridge Way SW, Seattle, WA 98106, USA, http://www.gqelectronicsllc.com/
2 DIYGeigerCounter https://sites.google.com/site/diygeigercounter/

Special Features of the Devices:

GMC-Devices:

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

Unique to these devices is that they can store a complete **History** of the observed counts over time**.** This History can be extracted and converted into files that can be treated just like the logging files.

Geiger counter functions (Speaker, Alarm, Saving mode, Date&Time, Calibration, Threshold), can be read from within GeigerLog, and some can also be set.

The GMC-Devices are connected via USB. The USB-Port used and its baud rate can be auto-discovered. All communication with the device is error-checked and corrected if possible.

RadMon+ Devices:

The RadMon+ device acts as an IoT (Internet of Things) device, and sends its data wirelessly to a special IoT server. GeigerLog expects to find the RadMon+ data on that server.

You have the option of using either a Geiger tube, or an environmental sensor for temperature, air-pressure, and humidity, or both. GeigerLog can handle up to all four variables.

The RadMon+ devices come as Do-It-Yourself kits; assembly is required, and the device needs to be configured. See more under Configuration on page 30.

In the default setting it is also assumed that the RadMon+ has both a working Geiger tube, as well as a working environmental sensor.

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

Quality Controls:

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

Data Files:

Several genuine and synthetic recordings of Geiger counter and environmental data are included, among them a recording from an international long-distance flight.

Availability

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

Introduction to GeigerLog

Installing and Starting GeigerLog

GeigerLog version **0.9.08** requires a **Python 3** environment. It will NOT run on Python2!

It was developed with Python version 3.5.2, and verified to run with Python 3.4, 3.6, and 3.7.

In addition to a Python3 environment a few Python modules are needed, which generally are not available in a default installation. Step-by-Step installation instructions for Python on **Linux** and **Windows** are provided in Appendix H – Installation on page 57.

The software

The software comes in a zipped package containing the Python scripts and associated resources like icons. The package is named `geigerlog-scripts-vXYZ.zip` (xyz is the version number).

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.

Installing

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

Starting the scripts

1. Make sure you have Python Version 3 installed, including all modules called in the scripts.
2. Start GeigerLog with:
   `/path/to/geigerlog/geigerlog`
3. If Python is not in your path, you may have to start GeigerLog with:
   `python /path/to/geigerlog/geigerlog`
   or: `python3 /path/to/geigerlog/geigerlog`

If that does not work, note any error message and look into Appendix H – Installation on page 57.

Look & Feel

The Python software depends on the host computer for the Look & Feel. If GeigerLog does not look the way you like it, see Appendix A – Look & Feel on page 40.

The GeigerLog Window

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

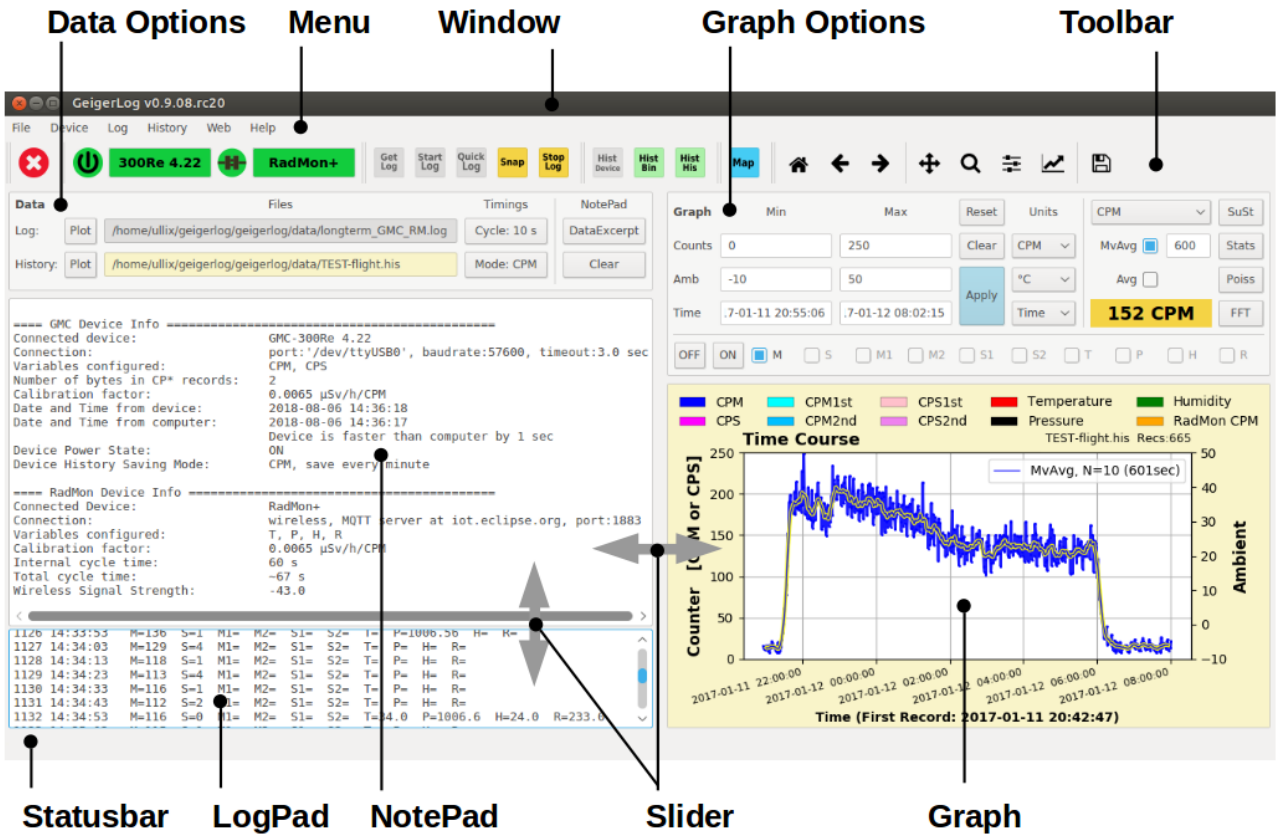


Fig. 1 GeigerLog Window with Annotations

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

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

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

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

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

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

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

Graph Dashboard: Settings to configure your graph.

Graph: Graphs will be shown here.

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

Quick Tour of GeigerLog

GeigerLog is best shown with a quick demo guiding you through typical usage steps. GeigerLog can be run without a connected 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 Time Course of CPM versus Time-of-Day.

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

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

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

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

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

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

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

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

Running GeigerLog

Connecting GeigerLog to Your Devices

To have GeigerLog interact with your device, you must establish a **connection** between them. Depending on the device, this could be based on a wire, like a USB cable, or it could be based on a wireless connection, like WLAN, also called WiFi in some regions of our planet.

Currently devices from the GMC Series and from the RadMon Series are supported. The GMC Series require a USB cable connection, while the RadMon Series require a network (LAN or WLAN) based connection.

Establishing a connection for the GMC Series devices may be more challenging than expected. And even though you may not own a RadMon device, you may get a connection to it anyway (explained below).

Before going into details, give it a try first. Start GeigerLog, and select menu **Device → Connect Device**.

Alternatively – and easier – click the red **Connect / Disconnect** button in the toolbar showing a symbolic open plug.

If successful, the Series buttons will turn green and show the devices detected, while the **Connect / Disconnect** button is now showing a closed plug on green background.

Note that you can **toggle the connection(s)** on / off using the red / green **Connect / Disconnect** button.

In addition to the button changes, a text will be printed into the NotePad area similar to this:

```
==== Connect GMC Device ========================================
Connected device:                 GMC-300Re 4.22
Connection:                       port:'/dev/ttyUSB0', baudrate:57600, timeout:3.0 sec
Variables configured:             CPM, CPS, CPM1st, CPM2nd, CPS1st, CPS2nd
Number of bytes in CP* records:   2
Geiger tube calibration factor:   0.0065 µSv/h/CPM
Geiger tube#2 calibration factor: 0.194 µSv/h/CPM
Date and Time from device:        2018-08-09 17:12:03
Date and Time from computer:      2018-08-09 17:12:01
                                  Device is faster than computer by 2 sec
Device Power State:               ON
Device History Saving Mode:       CPM, save every minute

==== Connect RadMon Device =====================================
Connected Device:                 RadMon+
Connection:                       wireless, MQTT server at iot.eclipse.org, port:1883
Folder                            'geigerlog'
Variables configured:             T, P, H, R
Geiger tube calibration factor:   0.0065 µSv/h/CPM
Internal count collection time:   60 s
Total cycle time (typical):       67 ... 70 s
Wireless signal strength:         -42.0 rssi
```

You can repeat these printouts any time by clicking on the buttons with the device name on them.

Connecting GMC - Devices

If you do see the left button in green with the model number of your GMC counter and the Firmware number on it, then you have made a successful connection and are set to continue. The brief info text in the NotePad area will come handy once you are familiar with GeigerLog and the counter.

If unsuccessful, a printout **in red** will tell you the reason. You will likely be advised to run **USB Autodiscovery** from menu **Help**.

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

When your USB-Autodiscovery ended successfully a dialog box will pop-up and tell you that it found a 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 its section **Serial Port** to make these just found settings the default.

For more complicated situations see Appendix B – Connecting GMC Device and Computer on page 41.

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

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 → GMC Series → Switch Power ON). GeigerLog's GMC device power icon

will change its state from Power OFF  to Power ON  .

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

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

Your GMC 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 have made the connection the correct Geiger counter model and firmware is shown!.

If this is not the case, then you may have to customize your model by modifying the configuration file `geigerlog.cfg` in its **Device** section. This is explained in Configuration on page 30.

Connecting RadMon+ - Devices

There will actually never be a connection between a RadMon+ device and GeigerLog, nor will the two ever talk to each other! That is the norm for IoT (Internet of Things [3]) devices.

The RadMon+ will be instructed to send its data to a server – in IoT lingo a broker [4]) – whenever it has data, and GeigerLog will be told which broker it is, and in which folder the data are saved. GeigerLog then connects to the broker and tells him which data it wants. The broker informs GeigerLog when new data are available, and GeigerLog downloads them.

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

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

4 A broker will be a server in your local LAN or anywhere on the internet, which runs MQTT software, https://en.wikipedia.org/wiki/MQTT . Tested servers are based on the Eclipse Mosquitto Open Source message broker Mosquitto https://mosquitto.org/ . You can easily install one on your own computer.

Operating your devices with GeigerLog

There are two operating modes in GeigerLog, **Logging** and **History**. The latter is only available for the GMC devices as it requires internal memory, which the RadMon+ devices don't have.

Logging means that GeigerLog gets fresh data from the devices, saves them in a log file, prints them on the screen, and plots them as a configurable graphic.

History means that GeigerLog reads data from the internal memory of a GMC Geiger counter, processes them, saves them as a file in a format matching the logging data, and plots them in the same way as the logging data as a configurable graphic.

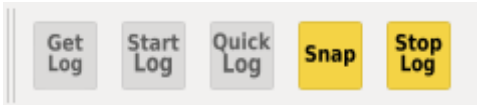
Logging

The process of logging is somewhat different between the GMC and RadMon+ devices, but to the user this is largely transparent. The following is applicable to both types of device.

In order to use a device for logging, it must be connected, and in the case of a GMC device it must also be powered on. This picture shows a RadMon+ being connected, as well as a GMC-300Re 4.22, and the latter is powered on (leftmost ON/OFF button is also green). So, both devices can be used for logging.

After the connection is made, some Device Info is printed into the NotePad area. If you want to see it again, simply click on the green buttons with the device name on them.

You further need to have a log file loaded, so click the **Get Log** icon in the toolbar to load an existing file, or define a new one. Now you can start by clicking the **Start Log** icon in the toolbar. The toolbar will change and while logging is ongoing, you can only stop the logging (Snap will be explained shortly). Other functions, which would interrupt logging, like exiting GeigerLog, are also disabled during logging.



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

Upon starting the log, GeigerLog queries all devices for all the variables they can provide (if the default is not what you want, it can be configured in the `geigerlog.cfg` file), reads them, prints them into the LogPad, saves them into the log file, and plots them.

This procedure is repeated after the cycle time has expired. This cycle time is set by clicking the **Timings Cycle** button in the **Data Dashboard**. A pop-up box allows you to enter a new cycle time of at least 0.1 seconds. A cycle time of less than 0.1 cannot be entered. (see Data Dashboard on page 25). With ongoing logging the cycle time cannot be changed!

Sometimes you may want to see fresh data right away and not wait for the next cycle. Simply click the **Snap** icon, and GeigerLog snaps a fresh record out of order and prints it into the NotePad area.

The result may look like this:

```
==== Manually Triggered Log Values ==============================
47 19:22:38 M=149 S=1 T=31.0 P=1000.29 H=41.0 R=244.0
```

To save printing space for better overview it uses the shortcuts M for GMC CPM, S for GMC CPS, T for Temperature, P for Pressure, H for Humidity, and R for RadMon+ CPM.

This record is also saved in the log file.

Logging with GMC - Devices

Any logging is strictly controlled by GeigerLog, **not** by the counter [5])!

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

The values always asked for by GeigerLog are:

CPM	: Counts Per Minute
CPS	: Counts Per Second

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

CPM	: Counts Per Minute as the sum of both tubes (makes no sense [6]))
CPS	: Counts Per Second as the sum of both tubes (makes no sense)
CPM1st	: Counts Per Minute for the 1st tube, the standard tube
CPM2nd	: Counts Per Minute for the 2nd tube, the low-sensitivity tube

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

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

CPS1st : Counts Per Second for the 1st tube, the standard tube
CPS2nd : Counts Per Second for the 2nd tube, the low-sensitivity tube

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

CPM = CPM1st = CPM2nd
CPS = CPS1st = CPS2nd

Logging with RadMon+ - Devices

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

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

My RadMon+ is set to collect counts from the Geiger tube for 60 seconds to determine a CPM value. Then the RadMon+ reads the data for temperature, air-pressure, and humidity from its BOSCH BME280 sensor. All 4 values are then sent through my wireless home network to my router and then to a broker server with address iot.eclipse.org, located in California, USA. The total of reading the 3 environmental variables, and processing and shipping all 4 variables takes an extra time of about 7 seconds. Then the RadMon+ starts a new cycle.

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

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

GeigerLog is currently pre-configured to access my own RadMon+ device upon clicking the **Connect / Disconnect** button, which will be successful when a connection to the broker has been made (I cannot guarantee that my RadMon+ will be always on, but there is a good chance for it).

Don't be surprised that the RadMon+ data of one GeigerLog cycle are found in different records in the logging file, as each data is sent by the RadMon+ individually, one after the other, and the GeigerLog cycle may cut in between!

A note on the logging cycle

The Geiger counter needs less than 1 ms (millisecond) [7]) 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

7 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.

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 Dashboard) for further smoothing the data.

This **oversampling** – sampling more often than really needed – has consequences for certain properties of the data, see Quality Control - FFT – FFT & Autocorrelation Analysis on page 35.

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

History

– only available on the GMC-Devices –

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

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

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

In theory you could download only a portion of the memory. But since this is laid out as a ring-buffer, you’d have to know very precisely what portion of the memory you want. Typically you won’t know this until after you have done the complete download and inspected the data. On top of this, a partial download may bring parsing problems (see Appendix E – GMC Device: Internal Memory, Storage Format and Parsing Strategy). So this is not an option.

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

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

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

NOTE: if you experience reading errors while downloading the history, or even a partially or completely unreadable *.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. The toolbar button **Hist Device** downloads the history from the counter to your computer, creates a plotable file, and shows the plot. Buttons **Hist Bin** and **Hist His** load files created by a previous history download, see more at Menu – History.

History Saving Mode

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

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

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

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 geigerlog.stdlog.
```

Of interest for debugging is the option '`-R`' (or '`--Redirect`'). While the program log file `geigerlog.proglog` has all output from GeigerLog, it does not have any error messages from Python itself, which are essential for debugging. With the redirect option another log file `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

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

Menu – File

Commands to plot 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)
- Show Summary Statistics (SuSt) — Print summary statistics for all data currently shown in the plot to the NotePad
- Show Statistics — In a pop-up window show detailed statistics for the data currently shown and selected in the plot
- Show Poisson Test — In a pop-up window show a 'Histogram with Poisson Fit' for the data currently shown and selected in the plot
- Show FFT & Autocorrelation — In a pop-up window show a 'FFT & Autocorrelation' analysis of the data currently shown and selected in the plot
- 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.
- Submenu: GMC-Series — Functions related to GMC-Devices
- Submenu: RadMon-Series — Functions related to RadMon Devices

Menu – Device – Submenu: GMC-Series

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

- Show Extended Info — Prints extended info about the device to the NotePad; see also Appendix D – The Device Configuration Meanings for some content included in info.
- Show Configuration Memory — Prints the device configuration as binary data in human readable format to the NotePad. Also see Appendix D – The GMC Device Configuration Meanings.
- Switch Power ON — Switches the device power ON (as if pressing the Power button on the device).
- Switch Power OFF — Switches the device power OFF (as if pressing the Power button on the device).
- Switch Alarm ON — Switches the device alarm ON.
- Switch Alarm OFF — Switches the device alarm OFF.
- Switch Speaker ON — Switches the device speaker ON.
- Switch Speaker OFF — Switches the device speaker OFF.
- Set History Saving Mode — Sets the device's mode of History saving. Can be:
  - OFF (no history saving)
  - CPS, save every second
  - CPM, save every minute
  - CPM, save hourly average
  - CPS, save every second if exceeding threshold
  - CPM, save every minute if exceeding threshold
- Set Date + Time — Synchronizes computer and device time by setting the device's date and time to the computer time.
- Reboot — Reboots the device.
- FACTORYRESET — Does a factory reset. Your device customization is lost, and the internal memory is cleared.

Menu – Device – Submenu: RadMon-Series

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

Menu – Log

Commands related to logging.

- 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
• Set Log Timings	The log cycle in seconds can be set in a pop-up window. The cycle time must be at least 0.1s; shorter times cannot be entered
• Start Logging	Starts logging. Requires that 1) a connection is made to the device, 2) the device is powered on, and 3) a log file is loaded The logged values will immediately be saved to the log file, printed to the LogPad, and plotted
• Stop Logging	Stops logging
• Quick Log	Start logging using the file default.log. The file will be emptied, before logging starts. If you want to continue logging into a previously selected default.log file, then choose Start Logging instead
• Add Comment to Log	Adds a comment to the log file; does not disturb logging or graphing.
• Show Log Data	Prints the log data to the NotePad.
• Show Log 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 Binary Data Excerpt — Prints the first and last lines of the data from the '.lst' 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 — Prints the data from the '.his' file to the NotePad
- Show History Data Excerpt — Prints the first and last few lines of 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 37.

Menu – Help

Some helpful information for running GeigerLog.

- Quickstart — A very short GeigerLog Manual
- GeigerLog Manual — Opens the GeigerLog Manual. Will attempt to open it locally, but if not available then does it online
- Devices' Firmware Bugs — Some info on firmware bugs perhaps of relevance to the user, and workarounds
- Radiation World Maps — A brief introduction into the use of the Radiation World Maps
- Occupational Radiation Limits — Info on occupational radiation limits of USA and Germany, and links for extended info.
- 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.
- About GeigerLog — A brief introduction to GeigerLog, as well as version and legal information

Toolbars

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

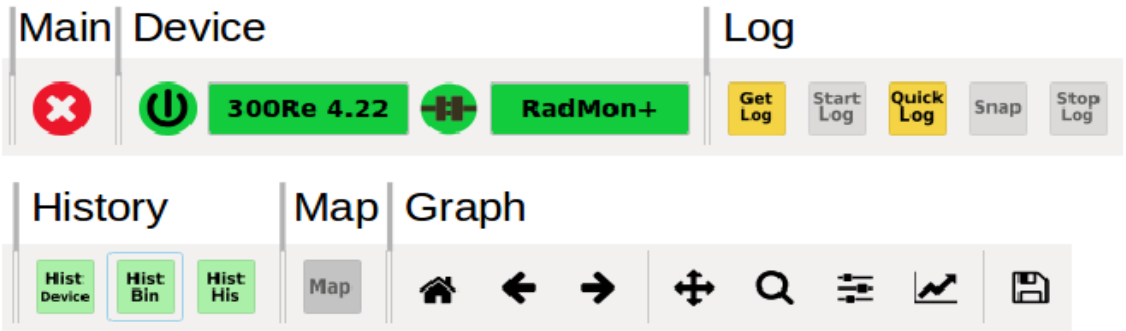


Fig. 2 The toolbars

The status in this picture is that both a GMC-series device and a RadMon+ series device are connected, and the GMC-Device is powered on. A log file is not loaded, and logging is not active.

- Main - Exit GeigerLog
- Device - Toggle Power of any connected GMC-device
  - Indicates which, if any, GMC-Series device is connected. In the figure a GMC-300E+ model is connected.
  - Toggle Connection to all devices.
  - Indicates which, if any, RadMon-Series device is connected. In the figure a RadMon+ device is connected.
- Log - Get a Log File
  - Start Logging
  - Start a Quick Log
  - Snap a single record during logging
  - Stop Logging
- History - Get History from device (possible only without ongoing logging; a GMC-Series device 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 37.
  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 Dashboard

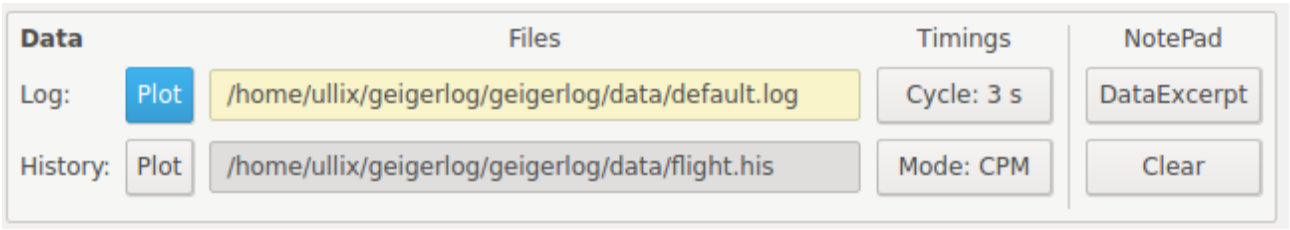


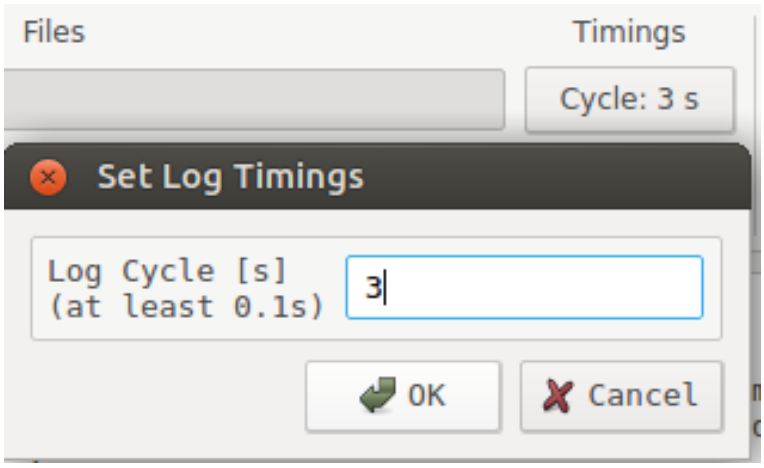
Fig. 3 Data Dashboard

Files

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.

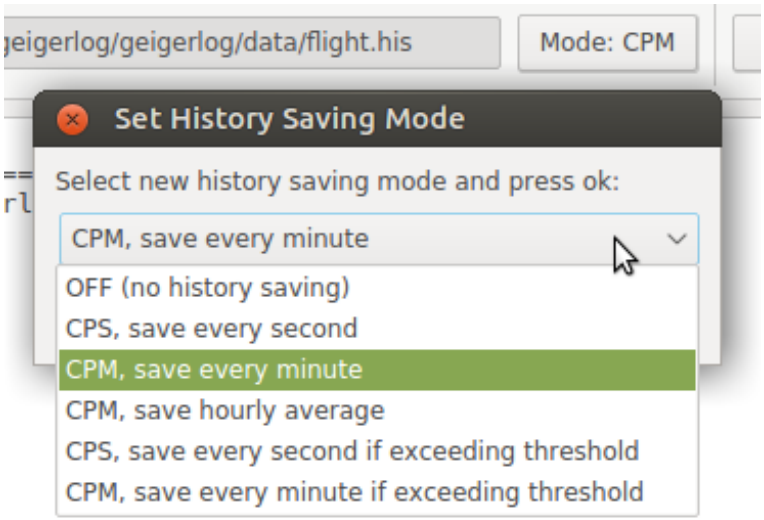
Timings

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



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

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



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

For an explanation of the options see Historyon page 16.

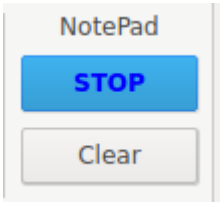
NotePad

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

The **Clear** button clears all content from the NotePad.

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

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



Graph Dashboard

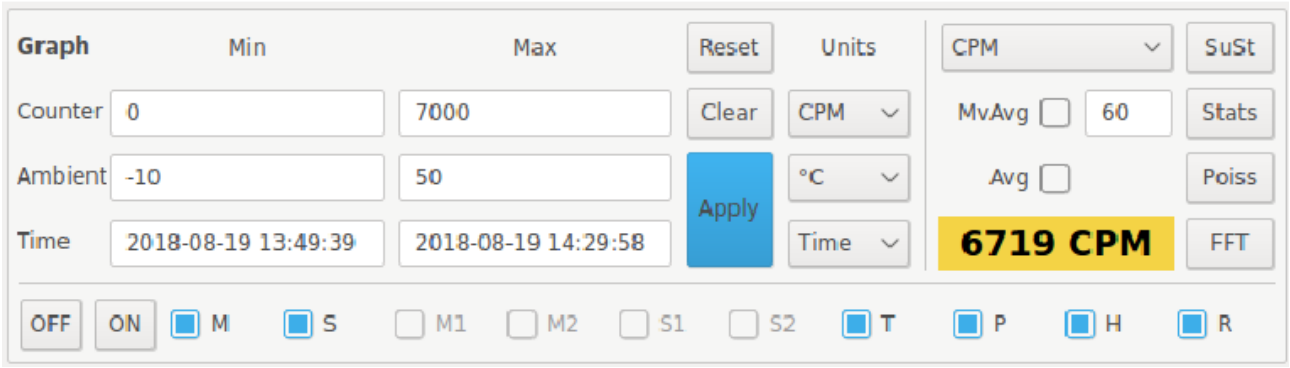


Fig. 4 Graph Dashboard

General

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

The graph is laid out as Time Course of your data versus time on the horizontal X-axis. It has two vertical axis:

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

Note: All data are displayed as recorded. The exception is air-pressure, which will be displayed as 'air-pressure minus 1000'. Since air-pressure is typically within the range of 970 … 1030 hPa, this transformation allows it to be displayed at the same scale as the other environmental variables. The values saved to the log file are always the full, original values.

What is displayed?

With up to 10 variables now available for display, it will often be important to reduce the number of variables displayed. The bottom row has buttons and checkboxes, which allow to show or hide a variable. Depending on active log or history file, not all variables may be available. The checkboxes of unavailable variables are grayed out and cannot be selected.

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

M	= CPM	from GMC-device
S	= CPS	from GMC-device
M1	= CPM1st tube	from GMC-device
M2	= CPM2nd tube	from GMC-device
S1	= CPS1st tube	from GMC-device
S2	= CPS2nd tube	from GMC-device

T	= Temperature	from RadMon-device
P	= Air pressure	from RadMon-device
H	= Humidity	from RadMon-device
R	= CPM	from RadMon-device

Min/max

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

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

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

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

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

Units

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

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

However, when µSv/h is used, the two curves should overlap! The µSv/h data are calculated from the calibration factors as built into the firmware of the Geiger counters [8]). E.g., for the first tube (M4011) of GMC-300 series as well as the GMC-500 series this is 0.0065 µSv/h/CPM. For the second tube SI3BG in the 500+ this number is 0.194µSv/h/CPM. However, that latter number is clearly wrong [9]).

This setting can be changed in **Device** section of the configuration file geigerlog.cfg. In GeigerLog's default configuration the calibration factor for the 2nd tube is set to 0.48µSv/h, based on the ratio to the M4011 in an experiment of my own [10]).

8 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 55.

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

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

Selected Variable

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

This selected variable will be highlighted in the graph with a brighter color and a thicker line, while the other variables will be dimmed. During logging, its last value will be shown in the Last Value area as black letters on a golden colored background (in the figure it shows '6719 CPM'). The other functions for analysis and quality control also work only on the selected variable.

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

If the selected variable is of the counter type, and these Poisson distributed data can be approximated by a Normal Distribution, two horizontal dashed lines will be drawn indicating the theoretical 95% range for the plotted data set, i.e. 95% of all data fall into this range, and 5% will be outside. If GeigerLog determines that the condition of Normal Distribution is **not** met, then **no** 95% range lines will be drawn, which is typically the case when the average is < 10 [11]).

If the checkbox **MvAvg** is checked then a Moving Average [12]) as a yellow framed line in the color of the selected variable will be plotted. The default duration for the moving average is 60 sec. E.g., with CPS data recorded once per second, applying a MvAvg of 60 sec will basically make a CPM curve out of it. For longer recording time moving averages over 600 or even 6000 may be appropriate.

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

Note: Sometimes, in particular when your log file contains very low and very high counts, it is advantageous to plot the data not in linear but in logarithmic scale. This can be achieved by clicking the Graph Toolbar icon labeled 'Edit axis, curve and image parameters' and then selecting Scale Log for the Counter-axis (you can also do further modifications via this toolbar icon).

The **SuSt**, **Stats**, **Poiss**, and **FFT** buttons are tools for the Quality Control of your data, see chapter Quality Control of your Data on page 32. The **SuSt** button prints a Summary Statistics of all variables currently displayed to the NotePad. The other three buttons act on the selected variable only and present their info in a pop-up-window.

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

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

Configuration

The hardware and the GeigerLog software may need configuration for certain situations.

Hardware – GMC Series

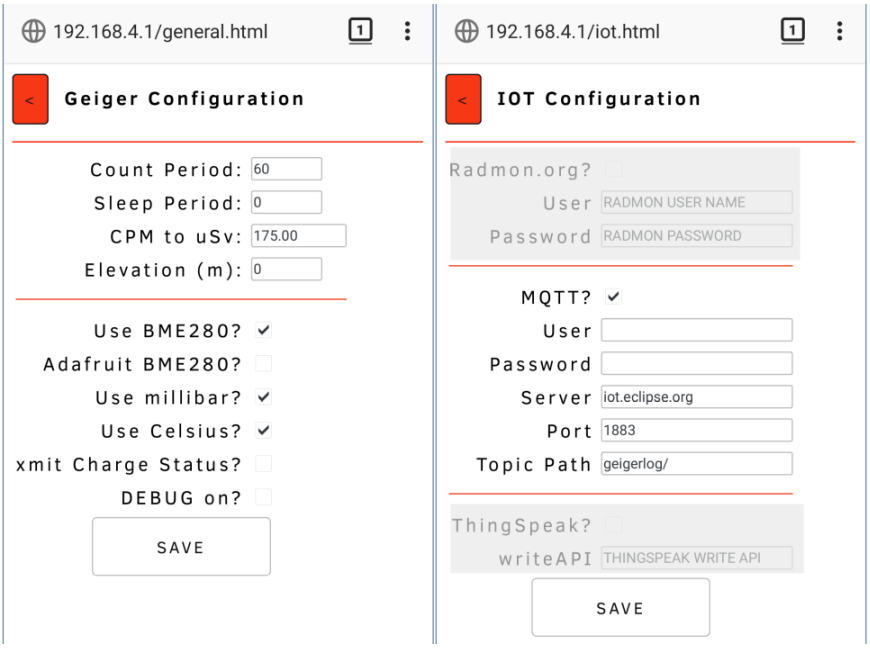
The GMC series counter actually do not need any configuration. A 'Factory-Reset' is recommended to be sure of a defined starting condition. All settings relevant to GeigerLog can be set from within GeigerLog.

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

Hardware – RadMon Series

The RadMon+ devices come as Do-It-Yourself kits; all parts – except the tube, power supply, and a case – are delivered, but you have to solder it yourself. Some basic skill in soldering is needed, but it is not overly difficult as there are no tiny SMD parts. You definitively want to have the manual [13]) ready when you do the assembly!

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



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

Note that the 'Topic Path' is 'geigerlog/' – ending with a slash '/'!

The same must be entered into GeigerLog's configuration file (see below) as 'RM-serverFolder = geigerlog/'!

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

Software

GeigerLog uses the configuration file `geigerlog.cfg`. The file is optional, but its content allows GeigerLog to handle those situations not covered by its internal defaults.

With the many different devices supported, some having model specific firmware bugs, changing firmware, new features and more, the config file may allow that an as yet unknown device can be configured so that it can be made to work with GeigerLog. Some of the firmware issues are laid out and explained in Appendix F – Firmware Differences on page 53.

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

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

The RadMon+ is not activated by default; it must first be defined in the configuration file. Here you can also activate the RadMon+ Demo-Mode.

Scaling

GeigerLog has received a powerful formula interpreter, to allow adjustments in support of environmental variables like temperature, air-pressure, and humidity. E.g., the temperature may need adjustments, which can be as simple as:

T_new = T_old – 0.2

The air-pressure is measured at the altitude of the location of your device, but typically for weather stations, you want it reduced to sea-level altitude. The formula is:

P_sealevel = P_altitude * (1 - (0.0065 * altitude) / (20 + 0.0065 * altitude + 273.15))**(-5.257)

Or, since it was recently discussed whether one needs a dead time correction for a Geiger counter. While I don't recommend doing it, because the counters have other problems that need to be solved first, if you want to use it the formula is :

CPM_true = CPM_observed / (1 - CPM_observed * 200 * 1E-6 / 60))

See the config file under topic 'Scaling' for details.

When this formula interpreter is used, the value of a variable is read, scaled by the interpreter, and saved as this modified value. The original value is not preserved.

Quality Control of your Data

The **SuSt**, **Stats** , **Poiss**, and **FFT** buttons in the Graph Dashboard area help you to check the quality of your data.

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

Also, a variable can only be used for analysis if that variable is shown in the plot. For the SuSt function all the variables in plot will be used. The other three functions use only one variable at a time, and it will be the selected variable, see Graph Dashboard on page 27.

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

SuSt – Summary Statistics

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

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

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

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

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

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

Stats – Statistics

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

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

Totals
  Filesize  =    311,786 Bytes
  Records   =        806

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

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

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

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

Poiss – Histogram with Poisson Fit

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

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

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

The value r^2 (in the graph as r2) is an indicator for the goodness of a fit. **A value of $r^2 \geq 0.9$ suggests a proper measurement.** If r^2 is smaller, then there may not be enough data points for a meaningful average, or some experimental error (source or counter shifted or removed during data collection ?) may have occurred.

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

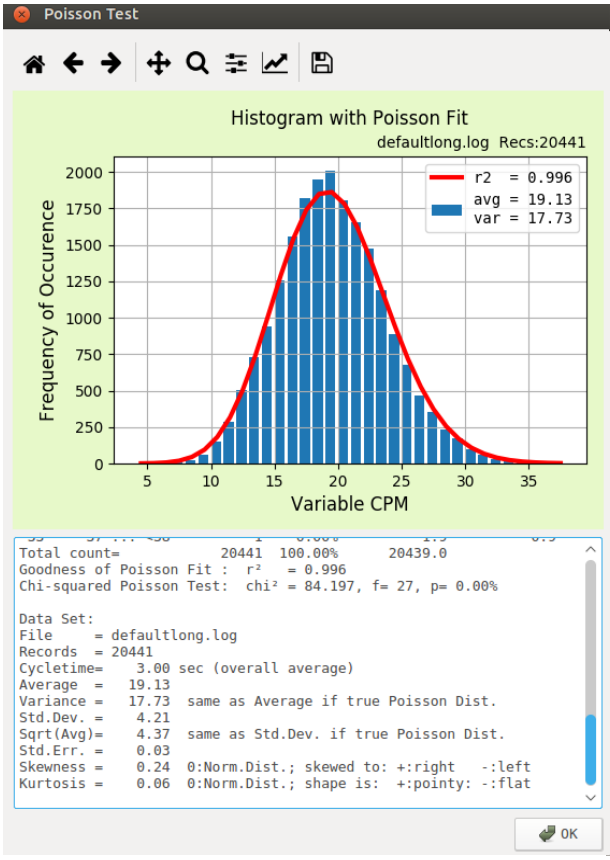


Fig. 5 Histogram of Low Count Rate

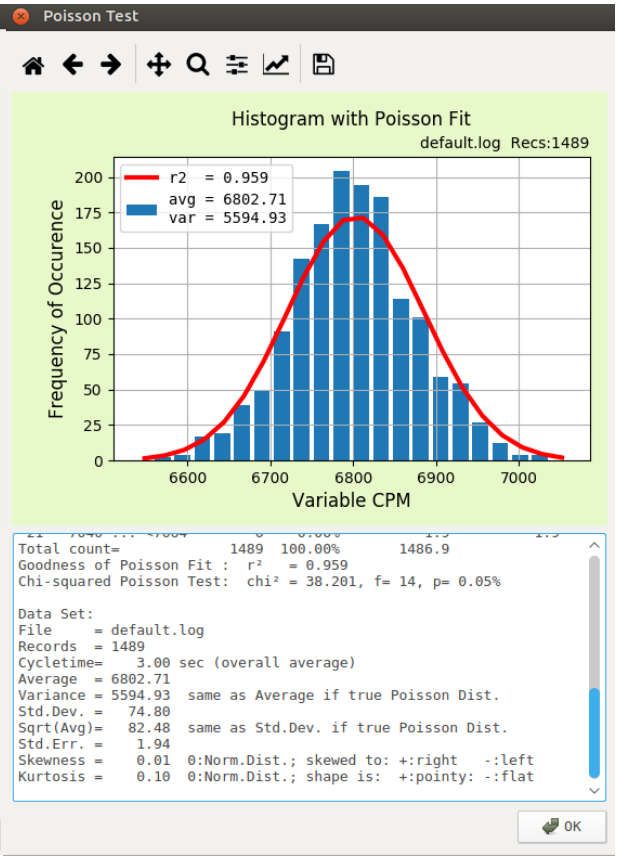


Fig. 6 Histogram of High Count Rate

FFT – FFT & Autocorrelation Analysis

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

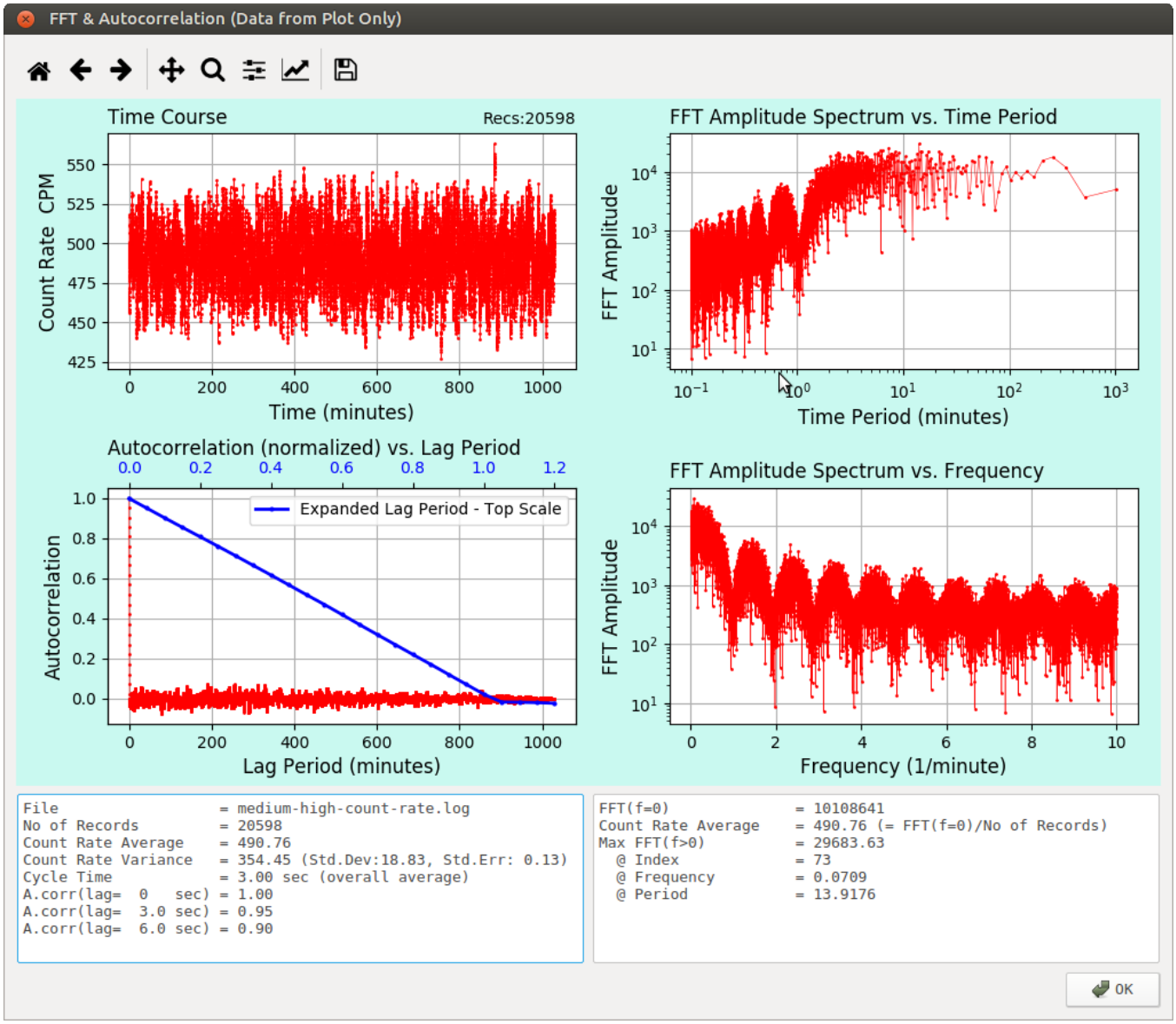


Fig. 7 FFT Analysis of Medium Count Rate Measurement

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

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

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

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.

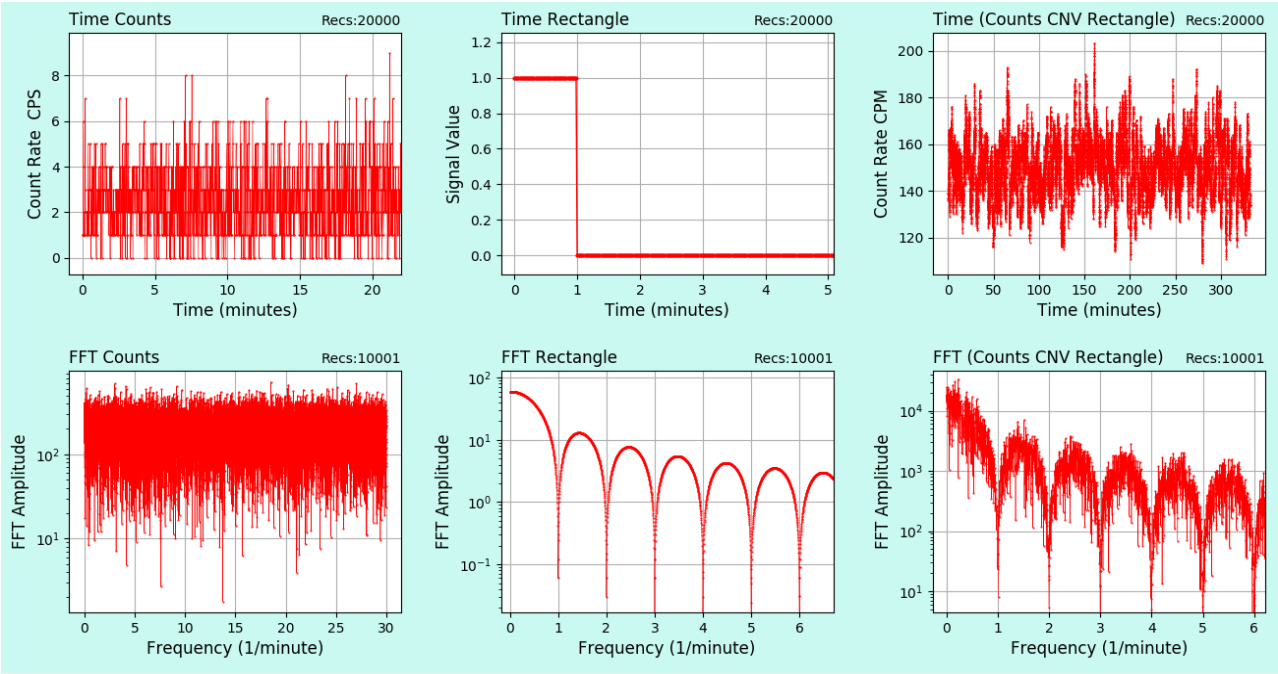


Fig. 8 Demonstration of the impact of convolution on an FFT spectrum (Synthetic data, CPS=2.5)

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

Radiation World Maps

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

Among the latter ones are:

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

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

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

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

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

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

A word of caution

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

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

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

Poor data will quickly destroy any value of those sites.

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

Unfortunately, Geiger counter readings fluctuate quite significantly. Thus when individual, single readings are posted, the values may be significantly higher or lower than the average, suggesting changes that don't exist. The fluctuation is largest at low count rates [14]), hence the reports of background rates are the most impacted: for a CPM=20 average background, 5% of the values can be expected to be greater than CPM=28 or smaller than CPM=10. That is almost a 3fold difference!

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

Governmental sites like this Swiss site [15]) provide only DAILY averages of quality controlled data!

Occupational Radiation Limits

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

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

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

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

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

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

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

Problems and Bugs

If your attempts to start GeigerLog fail, perhaps because the distribution you are using has different defaults, start GeigerLog from the terminal/command line, and look for error messages. Look through these error messages to find out if e.g. any modules are missing and what these modules are. Look through Appendix H – Installation on page 57 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 18.

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

The Python scripts use some resources which exist on your computer independently from Geiger-Log. 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 Devel → SysInfo.

Appendix B – Connecting GMC Device and Computer

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

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

The serial port, the baud rate and the driver

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

On Linux the driver is already part of the system. On Windows and Mac a driver must be installed.

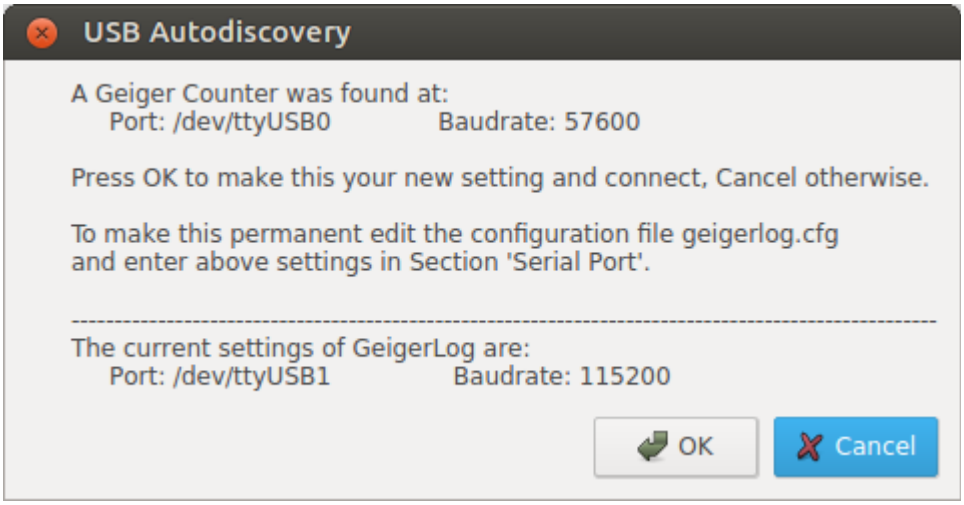


Fig. 9 USB Autodiscovery Pop-up Box

Drivers may be available for download on the GQ website [16]).

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.

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

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.

> **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.

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 GMC Device Configuration Meanings

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

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

GQ has recently disclosed the configuration of the 500 and 600, but some details remain unclear so far. (see: http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=4948), and at least 2 firmware bugs on the 500 series were discovered (see discussion in the topic). Please, report any problems via SourceForge (see Problems and Bugs on page 39).

The following list applies only to the 300 series:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

/
CFG_Save_DateTimeStamp, //this one uses 6 byte space

CFG_MaximumCFGBytes,

}EEPROMDATAT;

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

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

TOTAL_EEPROM_SAVE_TYPE

}HistoryDataFormatMarkingT;

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

```
SAVEOFF,

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

TotalSavedType

}SaveDataTypeT;
```

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

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

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

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

Date & Time Stamp

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

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

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

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

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

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

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

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

Overflow

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

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

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

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

Page Boundaries

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

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

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

After the page is deleted, all bytes up to the end of the page are set to 255 (in gray). The former time fragments 19 (min) and 12 (sec) become regular counts (in white) and the remainder of the Date&Time stamp beginning at 4098 (in yellow) has now become an ASCII tag with 85 bytes of supposed ASCII code following (only 3 bytes shown) [17)].

17 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.

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

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

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

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

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

The 255 value

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

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

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

```
./geigerlog keepFF
```

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

Correcting a Wrong History

It is an annoying procedure. The following is suggested:

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

Appendix F – Firmware Differences

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

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

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

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

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

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

History Download issues

The history is downloaded in pages of up to 4k (4096) bytes, which is hexadecimal 1000. The download is triggered by a request from the computer to send a page of the desired size. This de-

sired 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 GMC Device Configuration Meanings, page 45), 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 [18]):

```
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 en-

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

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

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

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

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

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

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

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

Appendix H – Installation

General

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

Python

This version of GeigerLog requires **Python version 3.x**. It has been developed on Python 3.5.2, and was confirmed to also work on Python 3.4, 3.6, and 3.7. The latest Python version is currently 3.7.0.

Pip

For any Python installation – be it on Linux, Windows, Mac, or else – it will be almost impossible to get a fully working installation without having the program **Pip** [19]) also installed! More on Pip is here: https://pip.pypa.io/en/stable/ .

All of the above Python versions come packaged with Pip, and while the Pip installation should be checked by default, make absolutely sure that it is and that Pip does get installed!

64 bit versus 32 bit

Use a 64 bit installation if your operating system supports it. The following assumes 64 bit for all downloads.

If you have to use 32 bit, find the equivalent downloads.

Administrative rights

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

Verify the current Python and Pip installation status on your machine

On your machine you may have installed only Python version 2.x (Py2), or only Python version 3.x (Py3), or both, or neither. Furthermore, depending on your operating system and distribution, as well as your history of 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 these commands entered in a Command Window:

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

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

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

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

Modules

In addition to a working Python Installation, you also need certain Python modules, which you probably won't have in a default installation. Installation instructions are below for these modules:

The qt4* modules will have to be installed via the distribution tools, like apt-get for Linux or specific installer packages for Windows. The version does not seem to be critical, but it must be a version matching your Python installation.

- qt4
- qt4-phonon (optional**)**

The qt4-phonon module has now been made optional, as there were too many problems with a Windows installation; no problems on Linux. Phonon had been used to give sound feedback. This is now replaced with a simple system sound on errors only. If you prefer to use Phonon you must activate it in the configuration file `geigerlog.cfg`.

The following modules are more easily installed using pip (latest versions as of Sept 2018):

- pip latest version: 18.0
- setuptools latest version: 40.2.0
- matplotlib latest version: 2.2.3
- numpy (MUST be at least version 1.14!) latest version: 1.15.1
- pyserial latest version: 3.4
- scipy latest version: 1.1.0
- paho-mqtt latest version: 1.4.0

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

Using pip

It is easier – and generally gives you a more recent version – to use the program Pip instead of the standard installation tool of your distribution. E.g., to do a fresh install of numpy, simply do from a Command Window:

```
pip3 install numpy
```

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

```
pip3 show numpy
```

If an upgrade is needed do it with:

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

To list all installed Python modules with their versions use:

```
pip3 list
```

If GeigerLog fails to run?

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

```
geigerlog  -dv
```

You'll find its output in the terminal and in the program log file `geigerlog.proglog`. Look through these messages to find out what – if any – modules GeigerLog misses. Modules may simply be not installed, but may be installed, though in a deprecated version.

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

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

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

before reinstalling with Pip. (Such was the case for the pyserial module in the Mint distribution.)

Installation of an Editor

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

GeigerLog has been developed on the editor Geany, which I do recommend. If you don't have a good editor yet, consider installing it. Latest release is Geany 1.33.

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

Linux - Installation

This was tested with **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 these commands:

```
sudo apt-get install python3-pyqt4
sudo apt-get install python3-pip
```
optional: `sudo apt-get install python3-pyqt4.phonon`

Using pip

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

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

```
sudo -H pip3 install pip –upgrade
sudo -H pip3 install setuptools --upgrade
sudo -H pip3 install pyserial --upgrade
sudo -H pip3 install numpy --upgrade
sudo -H pip3 install scipy --upgrade
sudo -H pip3 install matplotlib --upgrade
sudo -H pip3 install paho-mqtt --upgrade
```

Verify the numpy version:

```
pip3 show numpy
```

Remember that at least version 1.14 is required. Redo the numpy upgrade if needed:

```
sudo -H pip3 install numpy --upgrade
```

To review the installation:

```
pip3 list
```

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
```

or create a starter to be put in the panel.

Windows - Installation

This was tested with a fresh default installation of **Windows 10 Pro**. It is expected to also work on older versions of Windows, as it is using an older version of both Python 3 and PyQt4.

In contrast to this conservative approach I am also providing an **Alternative Windows Installation** for the most modern Python version, though that installation is a bit more challenging.

Verify the Python and Pip installation status on your Windows machine

On a fresh, default Windows installation there is no Python installed. Open a Command Prompt window and follow the steps in the "**General - Verify the current Python and Pip installation status on your machine**" chapter above. This will provide you with the situation on your system.

If you do have **Python 3** as **version 3.4.x (!)** already installed, make sure you have **Pip 3** also installed! If you have not, and you find no other way to add Pip, then de-install your current Python version and do a new install.

Likewise, if you have Python 3 installed, but in a different version, you may be able to find a matching PyQt4 version to install (and perhaps Pip), but likely it will be easier to just de-install your current version and do a new install.

Installation of Python 3.4.4

This version of Python was chosen because it is the last one with a Windows installer, and a matching PyQt4 package is available with also a Windows installer. Later versions provide source code only for at least one of the two.

Download from: https://www.python.org/downloads/release/python-344/ the Windows x86-64 MSI installer (for 64bit), which is the *.msi file python-3.4.4.amd64.msi, and install.

Attention: In the installer:

- make sure to select: "**Install for all users**"**!**
  If you don't, you face the additional hurdle in Windows (at least in Win 10Pro) that the installation will be deep under hidden directories, and you'll have to do tricks to add anything to it, like the PyQt4 package coming next!
- allow installation in folder **Python34**
- under customization:
  make sure that option **pip** *"Will be installed on local hard drive"*
  make sure that option **Add python.exe to Path** *"Will be installed on local hard drive"*

Installation of PyQt4

Download from: https://sourceforge.net/projects/pyqt/files/PyQt4/PyQt-4.11.4/ the 64bit file PyQt4-4.11.4-gpl-Py3.4-Qt4.8.7-x64.exe , and open **with administrative rights**.

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

Installation of Modules with Pip

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

The '`--upgrade`' (NOTE: **2** dashes before 'upgrade'!) at each command ensures that the most recent version of each module will be installed even if an older version is already installed.

With Pip for Py3 being installed, use it first to upgrade itself. **Only then** install the other modules.

```
pip3 install pip –upgrade
pip3 install setuptools --upgrade
pip3 install pyserial --upgrade
pip3 install numpy --upgrade
pip3 install scipy --upgrade
pip3 install matplotlib --upgrade
pip3 install paho-mqtt –upgrade
```

Verify the numpy version:

```
pip3 show numpy
```

Remember that at least version 1.14 is required. Redo the numpy upgrade if needed:

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

To review the installation:

```
pip3 list
```

Installation of GeigerLog

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

Start GeigerLog from a Command Prompt window with:

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

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

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

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

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

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

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

Alternative Windows - Installation

This provides you with the most up to date Python 3 installation, but requires the use of an unofficial repository plus some actions requiring a bit more familiarity with the Windows internals.

GeigerLog user Ikerrg is credited with providing the info for a PyQt4 installation from Wheel! He runs this installation successfully on Windows 7.

Cleanup

First, delete you current Python 3 installation, and remove all remnants, including those in the Windows-Registry. I used the freeware version of CCleaner (https://www.ccleaner.com/ccleaner) for it.

Installation of Python 3.7.0

This version is the very latest of Python, released only end of June 2018.

Download from: https://www.python.org/downloads/windows/ the Windows x86-64 executable installer (for 64bit), which is the *.exe file **python-3.7.0-amd64.exe**, and open **with administrative rights**.

Attention: In the installer:

- check: add Python to Path
- under Customize: check all options
- under Advanced Option:
  check: Install for all users
  check: add Python to environment variable
  the last two items (Download…) may remain unchecked
- confirm the install path c:\Program Files\Python37
- once finished: ignore the option to disable path length limit and close installer

Updating Pip and Installation of Modules

If your Command Prompt window is still open, then close it (Really!). Open again **as administrative user**. Remember to verify – see above – what your command is to call Python and Pip in their versions 3! I continue to use 'pip3' as a reminder to use the proper version, although if your system has no other Python version aboard except for the just installed one, your command will be only 'pip'. Type:

```
pip3 list
```

You will see only pip and setuptools listed (unless you have an additional Python installation) but both versions are outdated. First update Pip with (NOTE: **2** dashes before 'upgrade'!) :

```
python -m pip install --upgrade pip
```

You should now see Pip in version 18.0 or later. The upgrading of the other modules is easier.

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

```
pip3 install setuptools --upgrade
pip3 install pyserial --upgrade
pip3 install numpy --upgrade
pip3 install scipy --upgrade
pip3 install matplotlib --upgrade
pip3 install paho-mqtt –upgrade
```

Verify the numpy version:

```
pip3 show numpy
```

Remember that at least version 1.14 is required. Redo the numpy upgrade if needed:

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

To review the installation:

```
pip3 list
```

Installation of PyQt4

Here we use an unofficial repository to download a special version, a so-called 'Wheel' version.

Download from https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyqt4 the 64bit file for Python3.7.0: PyQt4-4.11.4-cp37-cp37m-win_amd64.whl. This wheel file allows installation with Pip!

In your Command Prompt window, opened **with administrative rights**, type:

```
pip3 install c:\path\to\PyQt4-4.11.4-cp37-cp37m-win_amd64.whl
```

Ooops, there is a sprocket missing in the wheel!

Indeed, there is a bug in the PyQt4 wheel, resulting in a missing file. GeigerLog will not start without it. Fortunately, we just installed this file elsewhere, and can copy it to where it also needs to be.

Find the file **`msvcp140.dll`** in the directory :

```
c:\Program Files\Python37\Lib\site-packages\matplotlib
```

(if package changes and file is not found here, you also find it under a Firefox installation folder!)

and copy & paste it into this directory:

```
c:\Program Files\Python37\Lib\site-packages\PyQt4
```

Installation of GeigerLog

Install as described for the first Windows installation, and GeigerLog should work!

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 verify the Python installation:

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

Now that you have Python working, verify your installation status on your machine with the commands given above under topic "**General - Verify the current Python and Pip installation status on your machine**"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

Once pip for Py3 is installed, use it to upgrade itself, and only then install the other modules.

The '`--upgrade`' (NOTE: **2** dashes before 'upgrade'!) at each command ensures that the most recent version of each module will be installed even if an older version is already installed:

```
pip3 install pip –upgrade
pip3 install setuptools --upgrade
pip3 install pyserial --upgrade
pip3 install numpy --upgrade
pip3 install scipy --upgrade
pip3 install matplotlib --upgrade
pip3 install paho-mqtt –upgrade
```

Verify the numpy version:

```
pip3 show numpy
```

Remember that at least version 1.14 is required. Redo the numpy upgrade if needed:

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

To review the installation:

```
pip3 list
```

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
```

Appendix I – License

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

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