GeigerLog
GLrelay Manual

by ullix

Addendum to GeigerLog
Version 1.5

March 2024

Intended Use of GeigerLog's GLrelay.py

Some GMC Geiger counters are equipped with WiFi, and allow to upload their values to a website. This is explained in the GeigerLog manual in chapters "**WiFi Equipped GMC Counter**" and "**Radiation World Maps**".

Sometimes, however, it is desirable to have the counter transmit the data by WiFi not to the net, but to a **local** target. Unfortunately, this is not easy due to a bug and a deficiency in GMC Geiger counters.

It is nevertheless made possible and easy by a workaround newly added to the suite of GeigerLog programs, the Python program **GLrelay.py**.

What is the basic problem?

The basic problem is that a GMC counter can send its data only to the HTTP port 80, which requires that the receiving software runs with full administrative rights. When you have a Web-Server, like Apache, running on your computer, it is indeed run with full administrative rights!

But, for security reasons, GeigerLog should **NEVER** be started as admin, instead always only as regular user.

Making things even more complicated, modern Web-Servers, like Apache, reject the GMC counter's request as a "security risk" because of certain issues in the way these requests are formulated by the counter's firmware. It can be overcome only by explicitly configuring the Web-Server as "**unsafe**". While you may find this acceptable in a local environment, it is definitely a No-Go proposal when you have outside-access to this computer!

Using Python, it is possible to overcome these hurdles without fiddling with big Web servers.

How does GLrelay.py work?

GLrelay.py acts as a middle-man between GMC counter and GeigerLog and corrects the deficiencies in the counter firmware calls. It is part of the GeigerLog suite, and is found in its **`gtools`** directory.

Its task is easy to describe: read the counter data from port 80, and write them – **unmodified** – to a port to which GeigerLog has access, like port 8000. This program intentionally cannot do anything else! Technically, GLrelay.py is a Web-Server on the receiving side, and a WiFi-Client on the sending side.

All three systems – GMC counter, GLrelay.py, and GeigerLog – need to use matching IP addresses and Port numbers. This is very simple when default configuration settings are used throughout, and will be discussed next. Non-default situations will be discussed later.

Special considerations for running Web-Servers

GLrelay.py is a Web-Server just like Apache, but with a much, much reduced feature level!

However, Web-Servers can never share port access! Thus, since GLrelay.py has to be listening on port 80 – as this is the only port the GMC counter can use – there cannot be any other Web-Server also listening on port 80!

But as port 80 has become the default port for any web traffic, it is thus basically impossible to run a second Web-Server. If you must do, then either stop it at least for the time you are running GLrelay.py (on e.g. Apache the commands are: `apachectl stop` and restarting with `apachectl start`) or reconfigure it to restrict its port access to exclude port 80. The latter, however, is a whole lot more difficult.

If stopping the second Web-Server is not an option, then another way is to run GLrelay.py on a second computer. A simple Raspberry Pi is more than sufficient. You will then need to follow instruction for the Non-Default configurations discussed later.

Since the purpose of GLrelay.py is to serve the restricted-access port 80, it is required that it is **STARTED WITH ADMINISTRATOR PRIVILEGES.**

The security risk is mitigated by allowing GLrelay.py to do only a single, well defined task, which is the relay of the counter data. It intentionally can't do anything else.

Default Configuration

This discusses the default configurations for a) GeigerLog, b) GLrelay.py, and c) the GMC counter, and presents results obtained with this setting.

a) Default Configuration – GeigerLog

GeigerLog's configuration is determined by settings in its configuration file **`geigerlog.cfg`**, found in the **`gconfig`** directory of the GeigerLog installation. It can be opened with a plain text editor.

GeigerLog can receive data by WiFi if its **WiFiClient device** is **activated** in the configuration file. Beyond activation make sure to set **`WiFiClientType=GMC`**, and leave all other settings on 'auto'.

b) Default Configuration – GLrelay.py

Actually, there isn't even a configuration needed for GLrelay.py.

GLrelay.py needs its **IP address** but finds it automatically. As it sits in a sub-folder of GeigerLog the default is to assume that both – GLrelay.py and GeigerLog – will be run on the same computer, so they both will have the same IP address!

Then you have to do nothing but start GLrelay.py, and your are done. But in this case you have to start it as **administrator** since you want access to the restricted port 80!

In **Windows** you have to open a Command Prompt Window as an administrator [1]). Then change into the GeigerLog directory and enter:

```
python gtools/GLrelay.py
```

In **Linux** it is done by opening a terminal and putting `sudo` in front of the command:

```
sudo gtools/GLrelay.py
```

After starting, GLrelay.py will reply with, as an example:

```
**********************************************************
GLrelay.py is configured for Listening at:  10.0.0.20:80
                         and Relaying to :  10.0.0.20:8000
```

In this example GLrelay.py has determined that it and GeigerLog are both installed on a computer which has the IP=10.0.0.20 in the local network. And it uses the default ports, which are port 80 for listening for data from the counter, and port 8000 for relaying the data to GeigerLog.

c) Default Configuration – Your GMC Counter

Assuming that your counter's WiFi is already switched to On, and the counter is connected to your local WiFi network, you only need to inform the counter where it should sent its data to.

Using the above data as example, go into the counter's menu to: **Server→Website** and enter: "10.0.0.20" (without the quotes). Next go to: **Server→URL** and enter: "GMC" (without the quotes). That's it!

Default Results

Logging from a GMC-500+ counter solely via WiFi – there is no additional USB connection between counter and GeigerLog – Figure 1 is what GeigerLog was getting with background radiation. GeigerLog is using its variable "CPM" to record data, which the counter is delivering as a value with the same name "CPM".

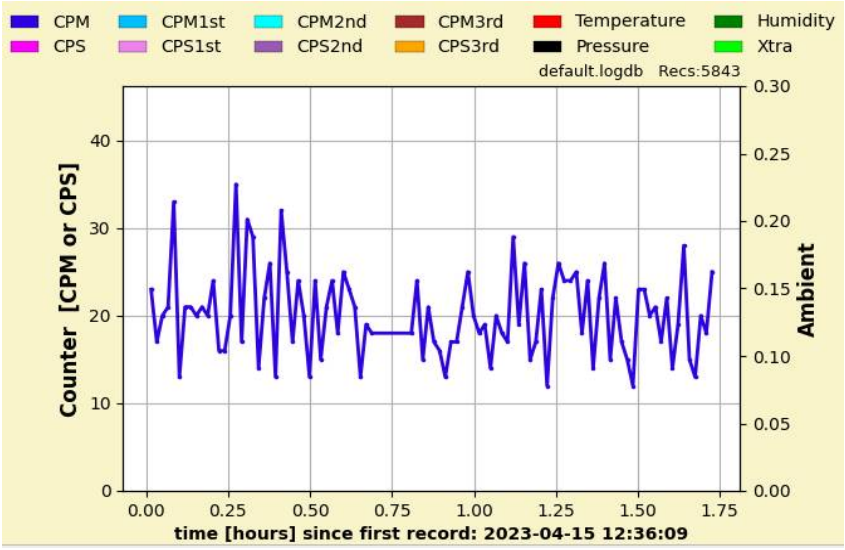

Figure 1: GeigerLog recording from a GMC-500+ counter via WiFi in All-Default settings

1 https://www.lifewire.com/how-to-open-command-prompt-2618089

Non-Default Configuration

Whenever you can, try the defaults first, before you start modifying. So you can confirm it works, before you move to a different situation, which may fail for reasons unrelated to the configuration settings, and you are problem searching at the wrong place.

All three systems, a) GeigerLog, b) GLrelay.py, and c) the GMC counter, allow certain modifications. Of course, all modifications have to be **consistent** among them!

a) Non-Default Configuration – GeigerLog

The possible settings in GeigerLog configuration file for a **WiFiClient device** are – as example – :

```
[WiFiClientDevice]
WiFiClientActivation = no
WiFiClientPort = auto
WiFiClientType = GMC
WiFiClientVariablesGENERIC = auto
WiFiClientVariablesGMC = CPM:CPM, CPM3rd:ACPM, Press:uSV
```

For a GMC counter all you can change is the WiFiClientPort, and the WiFiClientVariablesGMC.

The Port can be any number from 1024 ... 65535, default is 8000.

For the variables you have to map the GMC counter values CPM, ACPM, uSV to GeigerLog variables. You can chose any of the 12 GeigerLog variables, and map a counter value to it by putting a colon between the names **<GeigerLog variable name> : <counter value>**, like: CPM1st:CPM, which records the counter value CPM in the GeigerLog variable CPM1st.

Up to 3 GeigerLog variables can be set by joining each pair with a comma, like:

CPM1st:CPM, CPM2nd:ACPM, Xtra:uSV.

This records the counter value CPM in the GeigerLog variable CPM1st, the counter value ACPM in the GeigerLog variable CPM2nd, and the counter value uSV in the GeigerLog variable Xtra.

b) Non-Default Configuration – GLrelay.py

If you did change the **Port** in the GeigerLog config, then you need to account for this modification by starting GLrelay.py with an option. Allowed port numbers are 1014 … 65535 (incl.):

```
gtools/GLrelay.py --GLPort=<port number>
```

If you are running GLrelay.py on a computer different from the one GeigerLog is running on, then, while it knows the IP of the computer itself is running on, it has no way of knowing the **IP address of the GeigerLog computer**, and you must provide it:

```
gtools/GLrelay.py --GLIP=<IP address>
```

Or, if both Port and IP address are changed, then set both on the same command line:

```
gtools/GLrelay.py --GLPort=<port number> --GLIP=<IP address>
```

Should you forget anything, remember at least you can get **Help** with:

```
gtools/GLrelay.py -h   or    gtools/GLrelay.py --help
```

c) Non-Default Configuration – Your GMC Counter

Same procedure as with the default settings: Go into the counter's menu to: **Server → Website** and enter: "<new IP address>" (without the quotes and in the form of a typical IP address, like 1.2.3.4). Next go to: **Server → URL** and enter: "GMC" (without the quotes). That's it!

WiFi Equipped GMC Counters

GMC counters provide up to 3 values by WiFi, named: **CPM**, **ACPM**, **uSV** [2]).

You may have wondered why the default settings configure the use of the counter's CPM value only. The reason is that the other two are inappropriate or irrelevant. But, of course, if you want to record all three in GeigerLog, you can do so.

CPM is the instantaneous value of the counter's CPM.

ACPM is an averaged value of CPM; it is claimed to be calculated as the total number of counts accumulated by the counter since start (or reboot, reset) divided by the total expired time since that reset.

This is an unfortunate choice, as discussed in the forum [3]), because over time this value becomes less and less sensitive to changes!

uSV is calculated from CPM as CPM / 154 CPM / (µSv/h) [4]). So it brings no new information but is the very same as CPM multiplied with a fixed factor. You can easily do this in GeigerLog without having to record any additional data.

This is demonstrated in Figure 2 which shows a recording of all 3 values from a GMC counter.

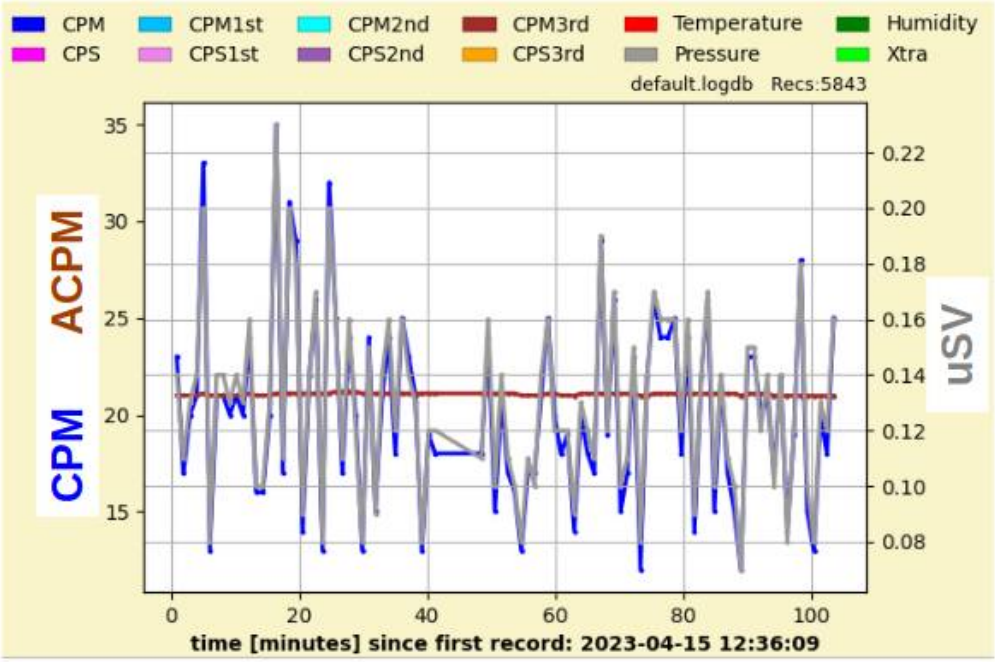

Figure 2: Recording of all 3 counter values in GeigerLog (left y-axis: CPM:dark blue, ACPM:brown, right y-axis: uSV:grey)

ACPM is the brown , nearly horizontal line, because the counter had been reset the day before. The blue line is the true CPM counts. It is almost completely hidden behind the uSV line. Make the calculation, the factor between uSV and CPM is a fixed factor of 154!

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

3 See e.g. Reply#4 in http://www.gqelectronicsllc.com/forum/topic.asp?TOPIC_ID=9815

4 For a counter with the M4011 tube; its sensitivity is 154 CPM / (µSv/h)

Using GeigerLog to make things easier

Using the counter directly to enter anything via its menu can create a lot of frustration given its poorly-responsive but happily-bouncing keys! It is easier done using GeigerLog.

Activate **GMC** in the GeigerLog config and start GeigerLog with the counter connected via USB cable. From the menu select **Device → GMC Series → Set GMC Configuration**. The dialogue shown in Figure 3 comes up.

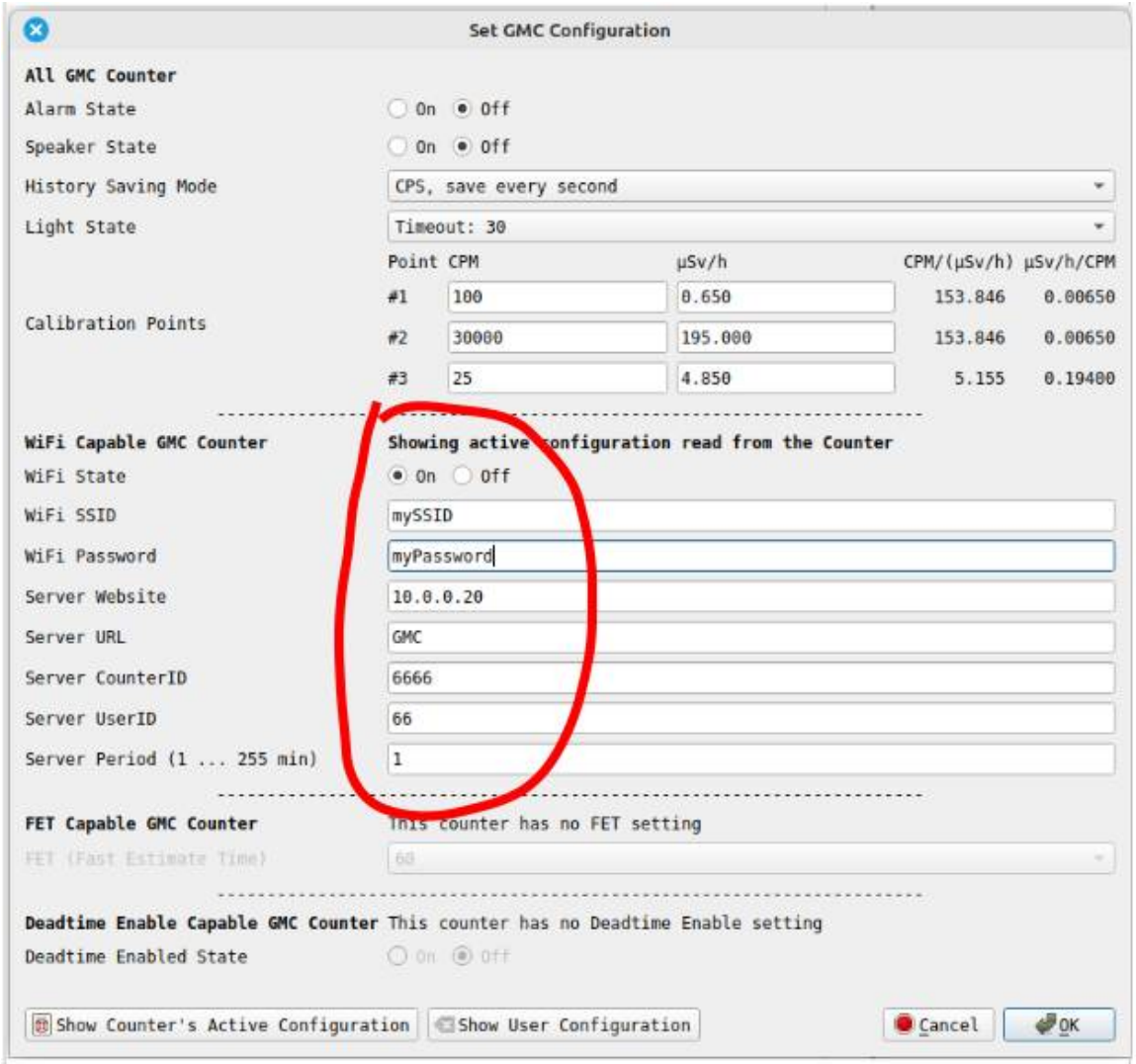

Figure 3: Settings for the GMC counter

You may already have entered SSID and Password for your local WiFi network. The “Server Website” is the IP address of the computer running **GLrelay.py,** not GeigerLog**!** However, mostly the two will be running on the same computer so then it is the same IP anyway.

The IP can be numeric (like: 10.0.0.20) or it can be text (like mycomp.local) if your network can resolve this. The “Server URL” **must** be “GMC” (without quotes).

The “Server Counter ID” and “Server User ID” can be anything; it is ignored by GeigerLog. The “Server Period” at its fastest setting is 1 (minute). You can only make it slower.

Is Simultaneous Use of a GMC counter by USB, Audio and WiFi Possible?

Of course. But to be sure – it is **NOT** necessary for any reason; each method of connection can be used alone, and none is made better by having any of the others connected too.

When GeigerLog listens to counts from **Audio** or **WiFi**, it has no idea who is delivering the counts. And if the devices were asked, they wouldn't answer, because they can't.

Only when GeigerLog is connected via **USB**(-To-Serial cable) can GeigerLog ask "who are you" and does get an answer. But even then it has no idea that Audio and WiFi are the very same guy!

Here is an example of a recording with a single GMC-500+ counter, using all three connection options:

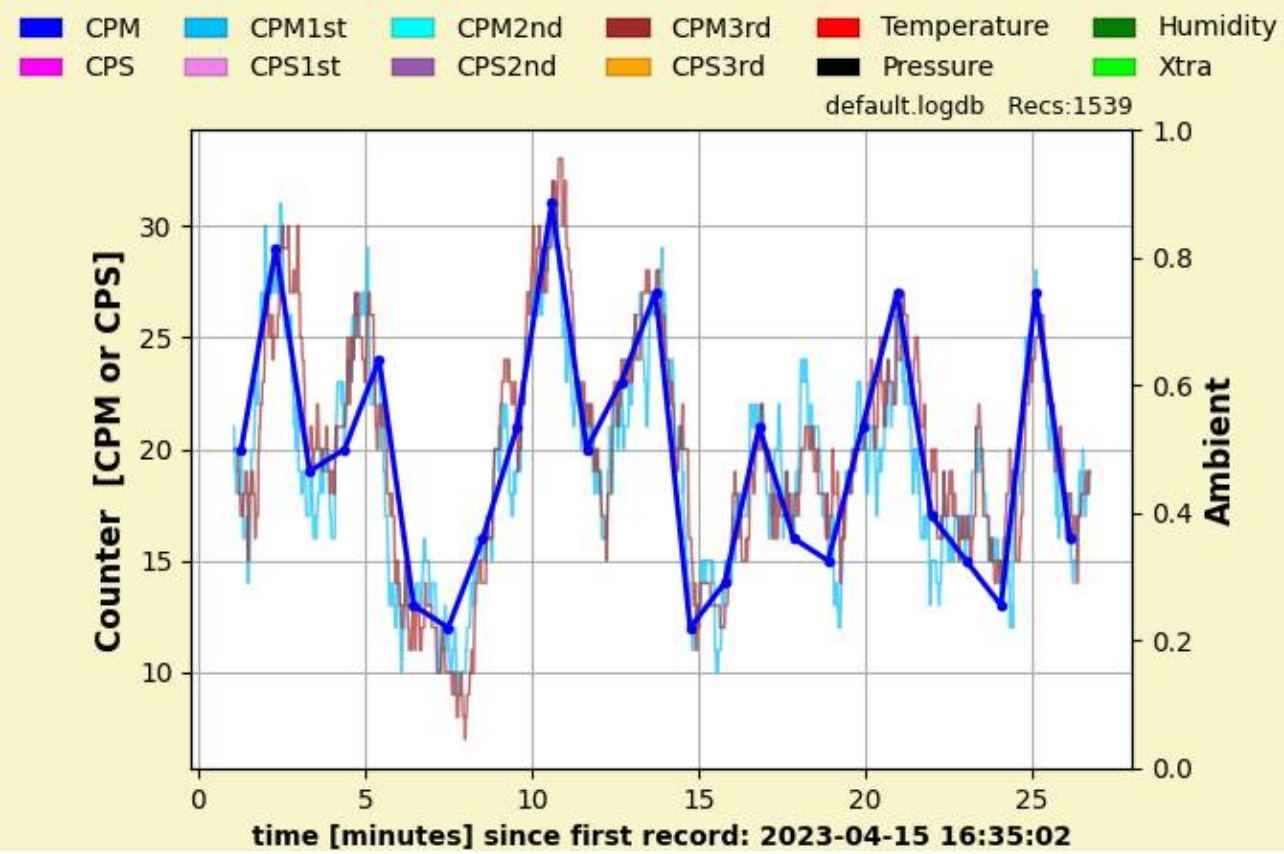

Figure 4: Using a single GMC-500+ counter, connected by USB (light blue), Audio (brown), and Wi'Fi (dark blue), showing CPM at background

As it should be, there is no difference between the methods except that the rate of recording is limited to once per minute with the WiFi, versus once per second with the two others.

ARCHIVE of OLD REDIRECTION METHOD

No longer recommended for use; the GLrelay.py method above is a lot more convenient!

Redirection for GMC counters

The GMC counters have a bug and a deficiency in the counter's firmware, which makes their use a lot more difficult than it has to be. Their use with GeigerLog requires a **URL redirection** with a **bad hack of an Apache Web server.** It is explained next.

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

Bug – "Unsafe" Operation Problem: The GMC counter's firmware forms the URL in such a way that modern servers reject their contact as a security issue. One can overcome this by explicitly putting an "Unsafe" into their configuration file [5]). But which Web server Admin would convert their server intentionally into a security risk?

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

Step 1: Enter "**HttpProtocolOptions Unsafe**" into the **`apache2.conf`** file (at the end is Ok) and restart Apache.

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

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

Note that the CURL module for php (look for: `php-curl`) must have been installed!
Note also the "dots" (".") in the code text – they are important!

```
<?php
$myIP = "10.0.0.20";   // Set your Forward-to IP
$myPort = "8000";      // Set your Forward-to Port
$myDevice = "GMC";   // Set for GMC counter
```

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

```
$GeigerLog_URI = "http://" . $myIP . ":" . $myPort . "/" . $myDevice .
$_SERVER['REQUEST_URI'];

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

Step 4: Run GeigerLog with its configuration setting `WiFiClientType=GMC`.

Error: Reference source not found shows a recent result running a GMC-500+ counter simultaneously with both Redirection and direct USB connection.