

## Online real-time Geiger counter to monitor radioactivity in air

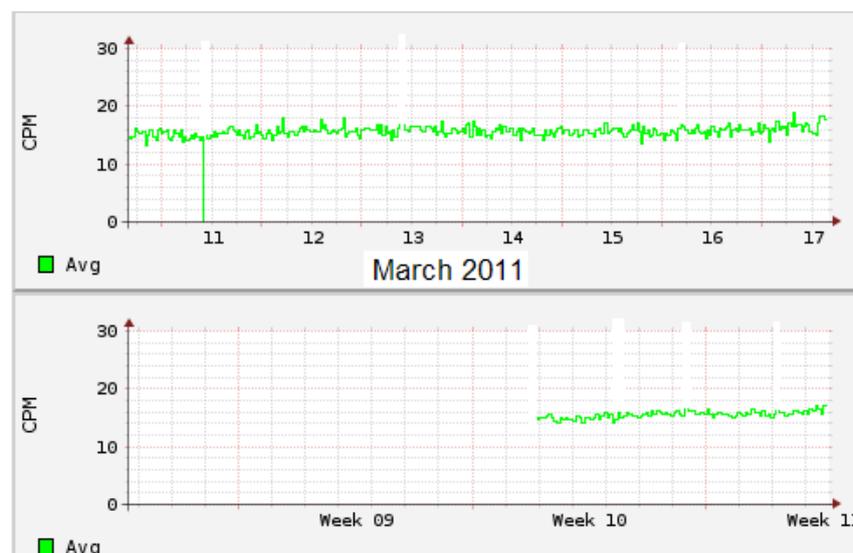
While living in Garching near Munich, Germany where is a research reactor I was curious to build a Geiger counter. I had a Russian made tube from 1970 in shelf for years and it appeared to be still functional.

Somehow just a month before March 13 2011 Fukushima nuclear catastrophe in Japan, I read a book [Radioactive boy scout David Hahn](#) and had assembled my counter. Intuition!?! In the following months I could not observe radiation background increase in Munich due to Fukushima. Counter was placed outside the building.

Could not see daily variations due to sunshine-solar radiation. During raining increased count rate, as rain washes down radioactive isotopes. This is normal. Counter tube wears-out after ca 6 month of continuous operation and count rate becomes sensitive to temperature. Counter was disconnected in April 2012 because nothing interesting happened. In Germany there is an official radiation monitoring network with a station in Garching:

<http://odlinfo.bfs.de/cvdata/091841192.php>

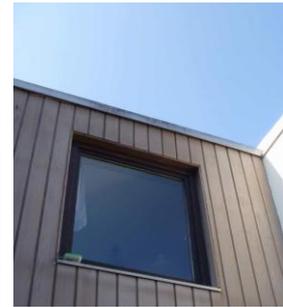
There is a high voltage part for Geiger tube and Arduino board as a click counter that is read out via USB link by a Debian webserver. Server reads out Arduino once in 5 min, inserts data in rrdtool database and from that generates data plots that are made available online.



The measured radiation background is ca 10-15 counts per minute. More when rain comes, at it washes out radioactive isotopes from air. No influence from Japan was visible. The counter tube run for 3 months continuously and afterwards had to be replaced. It started to generate many false clicks when the ambient temperature raised in the morning. It was not electromagnetic interference from neighbors shaving machine and not the moisture evaporation. Replacement of the Geiger tube helped. Looks like discharge-stopping gas (benzene vapor) that is contained in a small amount in the tube has been used up.

## Location of my counter

Hermetic food container for placement outside window.



## Some Geiger counter resources in Internet

- ✓ [List of Radiation Radioactive Nuclear Fallout Monitoring Websites](#)
- ✓ Commercial portable counter <http://www.terra-p.com/global/eng>
- ✓ Kits <http://www.best-geiger-counter.com/geiger-counter-kit.html>
- ✓ Home made counter schematics <http://www.techlib.com/science/geiger.html>
- ✓ Arduino interface <http://brohogan.blogspot.com/>

This tube is the same model as used by me.

Sensitive Geiger Counter Kit



Item Number : c6979  
Unit Price: \$89.95  
Quantity

[Refer this page to a friend](#)

**Detailed Description**

Sensitive geiger counter kit uses a Russian geiger mueller tube and is sensitive to Beta and Gamma rays. Want to know if that orange colored plate that was all the rage in the 60's is radioactive, how about your dad's radium dial watch, or the mantles on your camping lantern -this will tell you! Produces the familiar audio clicks and lights up a blue/green LED in response to radioactive sources. Random clicks are even heard from your kit as cosmic rays strike the tube! Uses a Russian GM tube that was made during the cold war (we test every one for proper operation), various resistors, capacitors, and a special miniature speaker. Operates from one 9V battery (not included). Comes with all parts, PC board, GM Tube and instructions (no case available). Size of PC board 4-5/16" x 1-1/2". Skill Level 2. Requires soldering.

## Counter hardware

Russian tube CTC-5 made in 1970 was used sensitive to beta and gamma radiation.

Simple setup, but HV pulser has no shutdown hence current drawn is ca 10 mA.

For long term operation use an external 9V power supply. Transformer windings numbers are adjusted during prototype assembly for correct HV range.

From speaker goes a 10 kOhm resistor to Arduino board which counts pulses.

Transistors are BC546, any NPN would be OK.

Transformer was wound on a 5 cm long ferrite rod piece from an old middle-wave radio receiver.

Closed loop toroid would be better because of less energy losses, but more hard to wind. Some people break toroid in 2 pieces, make winding and then glue together. It is worth trouble only if one goes for low consumption. Can also use ferrite core transformer.

High voltage winding was made first:

40 turns correspond to 9 V.  $x$  turns correspond to 400V.

From this proportion  $x=40 * 400 / 9 = 1777$  turns.

In practice I made HV winding of 1500 turns, diameter 0.1 mm layer wise isolated by tesa film.

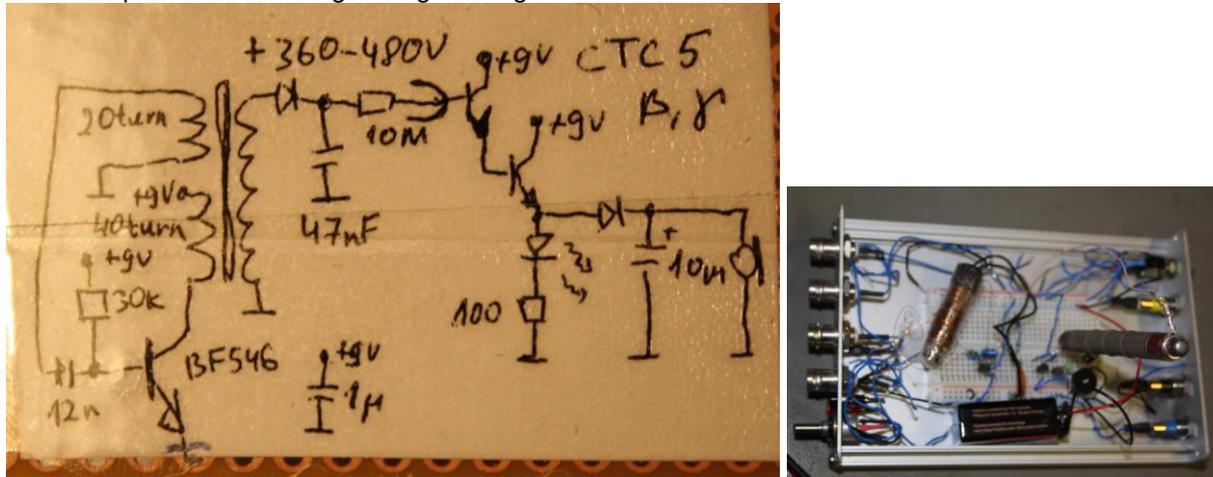
Then put 20 turns for transistor base driving isolated by tesa film.

And, finally, put ca 50 turns of thicker wire diameter 0.25 mm - 0.5 mm for the driving side.

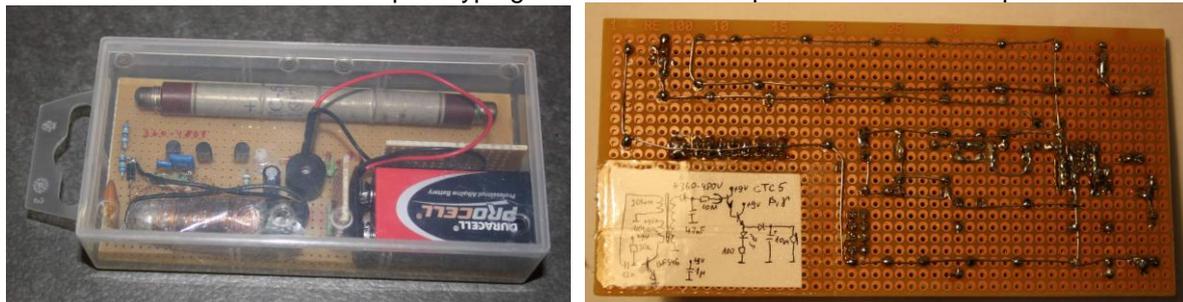
Next step is to get HV part running. Might be necessary to do trial and error and swap begin and end of transistor base driving coil.

Last step is to measure HV. And remove some windings until the optimal voltage for the counter is obtained. Did it using oscilloscope HV divider. Ca 40 turns were left in my case. Turns could also be adjusted for 5 V USB power supply.

Present circuit consumes ca 10 mA current. Some more advanced circuits measure HV and shut down the pulser when voltage is high enough. Such circuit would use less current.



Circuit was first assembled on a prototyping breadboard. Later portable version in a plastic box.



## Pulse counter using ATMEL Arduino

From speaker signal is sent to pin 3 of Arduino via 10kOhm resistor.

Arduino simple pushbutton programm example is used. (Need to change to long int for avoiding overflow). It sends serial data with click number.

```
// Geiger

int i=0;
int SW1=3;
int LED=13;
int TOG=0;
byte value_1, value_2=0;

void setup()
{
  Serial.begin(9600);
  pinMode(SW1,INPUT);
  digitalWrite(SW1,HIGH);
  pinMode(LED,OUTPUT);
  Serial.println("GEIGER");
}

void loop()
{
  value_1=digitalRead(SW1);
  if(value_1)
  {
    // delay(50);
    value_2=digitalRead(SW1);
    if(value_2)
    {
      i++;
      Serial.println(i,DEC);
      if((TOG!=0)TOG=0;else TOG=1;
      digitalWrite(LED,TOG);
      do{
        }while(!digitalRead(SW1));

    }
  }
}
```

## Debian scripts

Linux machine reads ttyUSB0 port and makes jpg plots using rrdtool. Plots are saved as images and displayed in Webpage.

```
rrdtool create geiger.rrd --step 300 \  
DS:geiger:COUNTER:600:U:U \  
RRA:AVERAGE:0.5:1:2016 \  
RRA:AVERAGE:0.5:6:1344 \  
RRA:AVERAGE:0.5:24:2190 \  
RRA:AVERAGE:0.5:144:3650 \  

```

### Main programm

```
A=$(cat /dev/ttyUSB0 |head -n 1);  
echo $A;  
printf " ";  
A=$((A*60));  
echo $A  
printf " ";  
rrdtool update /var/www/geiger.rrd N:$A;  
#rrdtool fetch /var/www/geiger.rrd AVERAGE;
```

```
rrdtool graph /var/www/geiger.png --start -1d --end now --vertical-label "CPM" $  
--upper-limit 30 --lower-limit 0 --rigid \  
DEF:average=/var/www/geiger.rrd:geiger:AVERAGE LINE1:average#00FF00:"Avg" \  

```

```
rrdtool graph /var/www/geiger1.png --start -1w --end now --vertical-label "CPM" $  
--upper-limit 30 --lower-limit 0 --rigid \  
DEF:average=/var/www/geiger.rrd:geiger:AVERAGE LINE1:average#00FF00:"Avg" \  

```

### This code is executed every 5 min using crontab

```
alix1dja:/usr/bin# crontab -l
```

```
*/5 * * * * geiger.sh
```

## Arduino program uses interrupts for counting Geiger counter clicks

```
=====

int ca0=0, cb0=0, cc0=0;
int state = LOW;

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  attachInterrupt(0, stateChange, CHANGE);
}

void loop()
{
  Serial.println(cc0);
  if (ca0 != cb0){cc0++;cb0=ca0;}
  delay(100);
}

void stateChange(){state=!state; ca0++; digitalWrite(13, state);}

=====
```

**This program for Arduino uses interrupt for counting Geiger counter clicks and meanwhile Arduino can do other things too.**

Serial data are sent showing analog input values. One can send command from Linux PC to switch on a relay that can switch on/off some 220V load. Another command is for watchdog relay. If Watchdog variable is not zeroed, another relay removes power from computer.

```
=====
long i=0;
String relay="OFF";
int ca0=0, cb0=0, ca1=0, cb1=0;
long cc0=0, cc1=0;
int state = LOW;

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  digitalWrite(11, LOW);
  attachInterrupt(0, stateChange0, CHANGE);
  attachInterrupt(1, stateChange1, CHANGE);
}
void stateChange0(){state=!state; ca0++; digitalWrite(13, state);}
void stateChange1(){ca1++;}

void loop() {
  String kommand="";
  String report="";

  for (int x=1; x<20;x++){
    if (ca0 != cb0){cc0++; cb0=ca0;}
    if (ca1 != cb1){cc1++; cb1=ca1;}
    delay(20);
  }
  while (Serial.available())
  {
    char ch = Serial.read();
    if (ch == 'C')
    {
      for (int x=1; x<8;x++)
      {
        if (Serial.available())
        {
          char ch = Serial.read();
          kommand = kommand + ch;
        }
      }
    }
  }
  if (kommand=="ommand0") {relay="OFF"; digitalWrite(13, LOW); digitalWrite(11, LOW);}
  if (kommand=="ommand1") {relay="ON"; digitalWrite(13, HIGH); digitalWrite(11, HIGH);}
  if (kommand=="ommandW") i=0;
  if (kommand=="ommandR") i=1990;
  report = report+analogRead(A0)+" "+analogRead(A1)+" "+analogRead(A2)+" "+analogRead(A3)+"
"+analogRead(A4)+" "+analogRead(A5)+" "+i+" "+relay+" "+cc0+" "+cc1;
  Serial.println(report);

  if (i>2000) {
    i=0;
  }
}
```

```
digitalWrite(12, HIGH);  
delay(5000);  
digitalWrite(12, LOW);  
}
```

```
i++;
```

```
}
```

=====