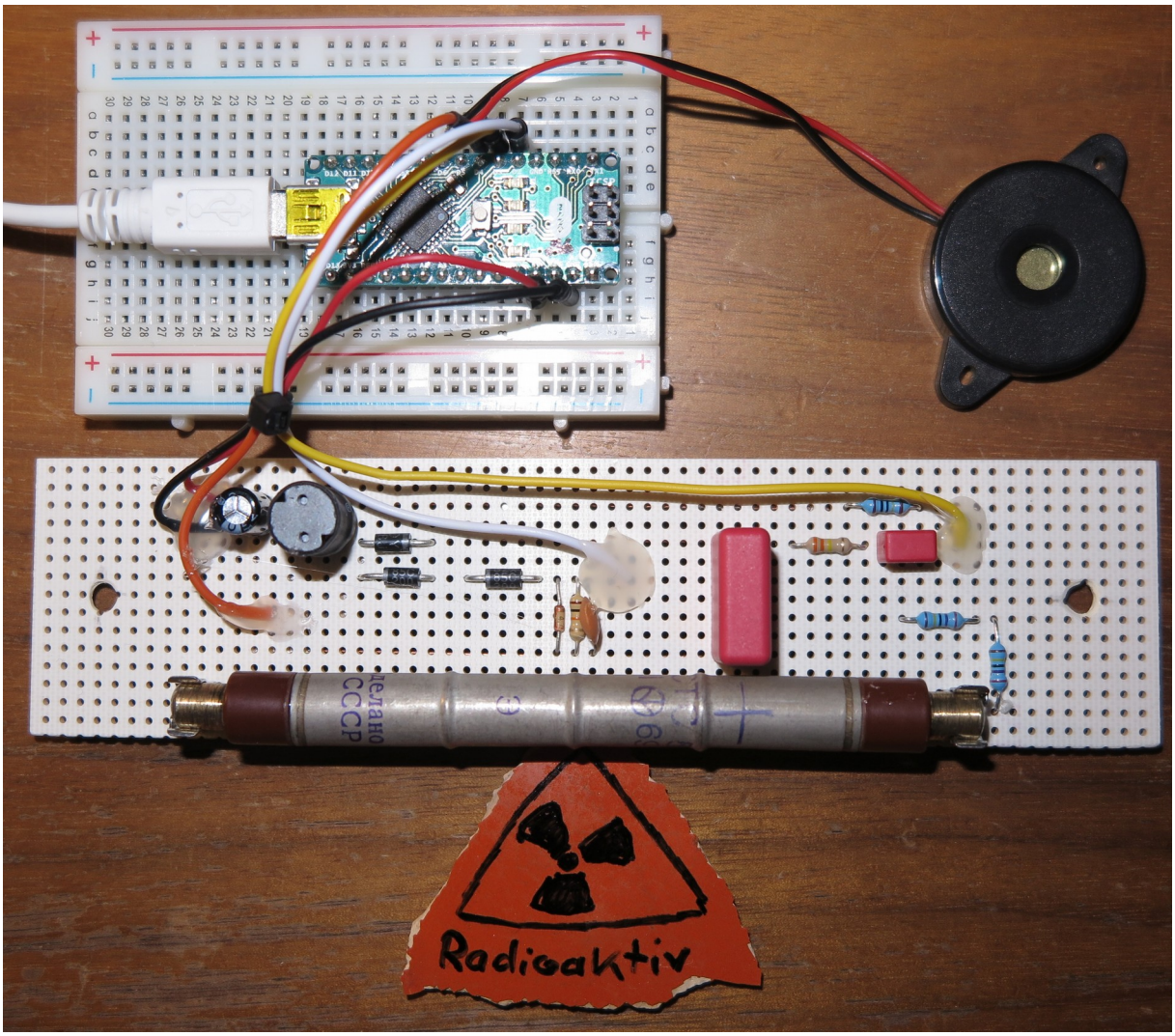
Simple-Arduino-Geiger



Simple-Arduino-Geiger auf einer Punkt-Streifenrasterplatine,
Im Hintergrund der Arduino Nano auf einem Steckbrett mit den Piezo-Lautsprecher,
im Vordergrund der Test-Strahler (Keramik-Scherbe mit Uran-Lasur)

Hier wird beschrieben wie man mit relativ wenigen Bauteilen und einem Arduino-Board einen Geigerzähler aufbauen kann. Die gemessenen Daten können über die (virtuelle) Serielle Schnittstelle (über USB) an den Rechner übertragen und angezeigt werden. Des weiteren erzeugt ein Piezo-Lautsprecher das charakteristische "Geticke".

Der Phantasie sind aber keine Grenzen gesetzt und viele Einsatzgebiete sind möglich. Die gemessenen Daten können vom Arduino weiterverarbeitet werden - z. B. für eine Anzeige, eine Datenübertragung ins Internet, Datenlogging auf SD-Karte oder Alarmierung bei Grenzwertüberschreitung.

Eigenschaften der Schaltung:

- Preiswert
- Präzise
- Einfach aufzubauen
- Gut verständliches Prinzip
- Extrem geringer Stromverbrauch
- Gut lötbare Teile (keine kleinen SMD-Bauelemente)

Viel Spaß beim Nachbauen!

Versionsgeschichte

2018_12_07	Jürgen Böhringer	Initiale Version
2018_12_17	Jürgen Böhringer	Neue Bilder, Layout für Punkt-Streifenrasterplatine

Hintergrund

Vermutlich gibt es kaum eine Messgröße, die für große Diskussionen sorgt wie diese: RADIOAKTIVITÄT.

26.04.1986: In ukrainischen Tschernobyl explodiert der Reaktor Nr. 4. Die Katastrophe wird von der Sowjetunion zunächst verheimlicht. Allerdings detektieren westliche Geigerzähler im schwedischen Forsmark den radioaktiven Fallout. Zunächst befürchtet man dort einen Störfall am eigenen Atomkraftwerk. Aber langsam wird klar welch große Katastrophe sich ereignet hat: In der Ukraine, Weißrussland und Russland müssen ganze Landstriche dauerhaft evakuiert werden, in Deutschland sind große Teile der Ernte verstrahlt und müssen vernichtet werden. Kinder sollen nicht mehr im Sand spielen. Die Regierung (CDU) wird nicht müde zu verkünden dass in Deutschland alles harmlos sei. Jedoch schätzen viele Wissenschaftler die Bedrohung anders ein und warnen vor der Strahlenbelastung über direkte Strahlung und die Nahrung. Die Bürger sind verunsichert.

Für mich, damals angehender Elektronikbastler, war es natürlich naheliegend einen Geigerzähler zu bauen. In einer Elektronik-Zeitschrift war auch bald eine Bau-Anleitung gefunden: Ein Zählrohr, eine Topfspule als Transformator bewickelt, ein paar Transistoren, Widerstände und Kondensatoren. In Katalogen suchte ich nach den Bauteilen. Hier musste ich allerdings bald ernüchtert feststellen dass das Zählrohr 200.- DM (Deutsche Mark - heute inflationsbereinigt ca. 200.- €) kostete. Hieran scheiterte das Projekt. Aber abgesehen davon wäre ich vermutlich damals, als Elektronik-Anfänger, auch an der Inbetriebnahme der Schaltung gescheitert.

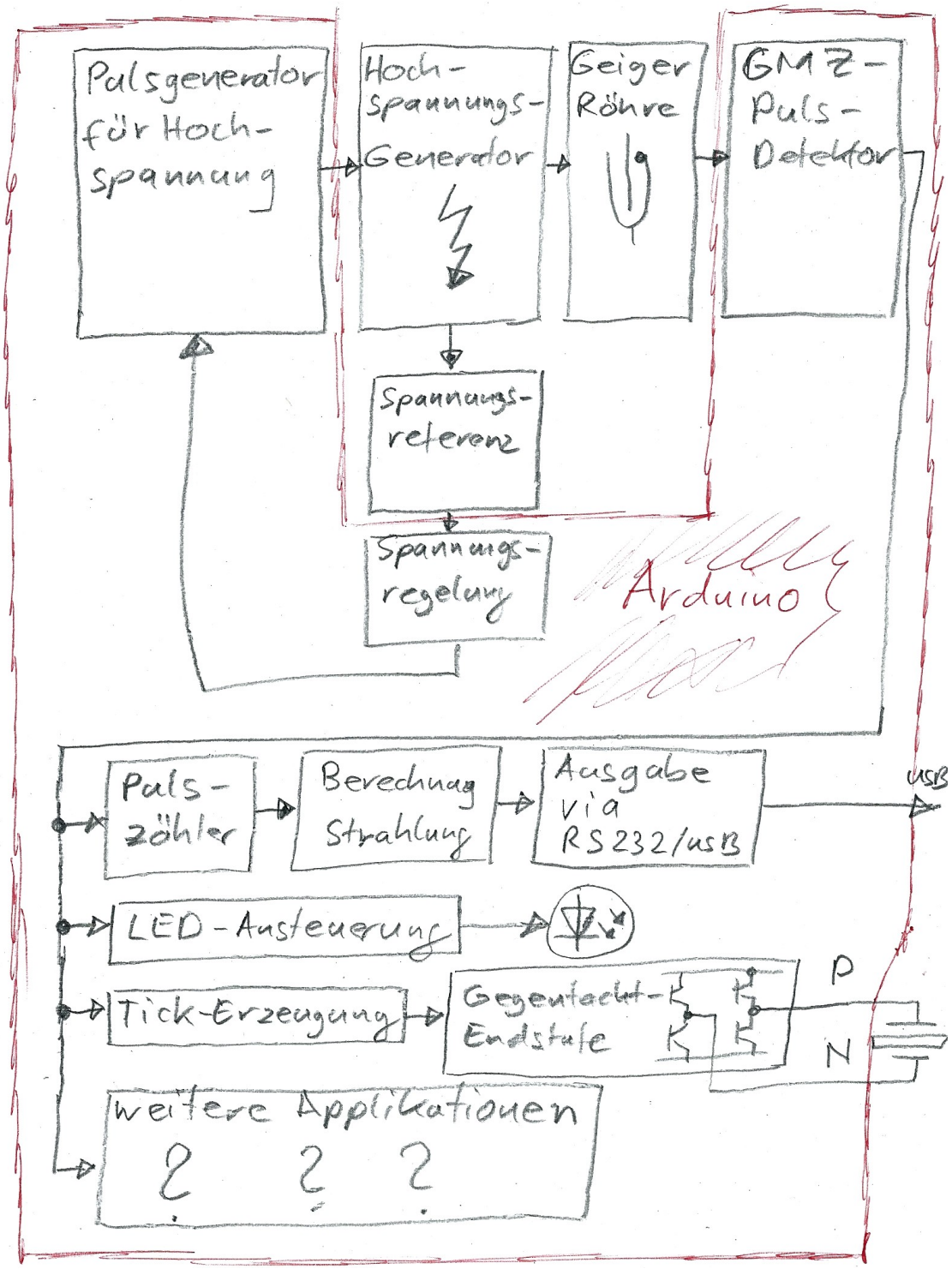
Radioaktivität zu messen ist unverändert interessant. Denn: Unfälle können jederzeit passieren - in AKWs, Endlagern und bei Transporten.

Heute ist es wesentlich einfacher einen Geigerzähler zu basteln: Zum einen gibt es bereits bei EBAY für rund 13.- € ein recht empfindliches SBM-20-Zählrohr, zum anderen kann man, mit Hilfe eines Mikrocontrollers die Schaltung sehr einfach, übersichtlich und preiswert halten.

Schaltungsbeschreibung

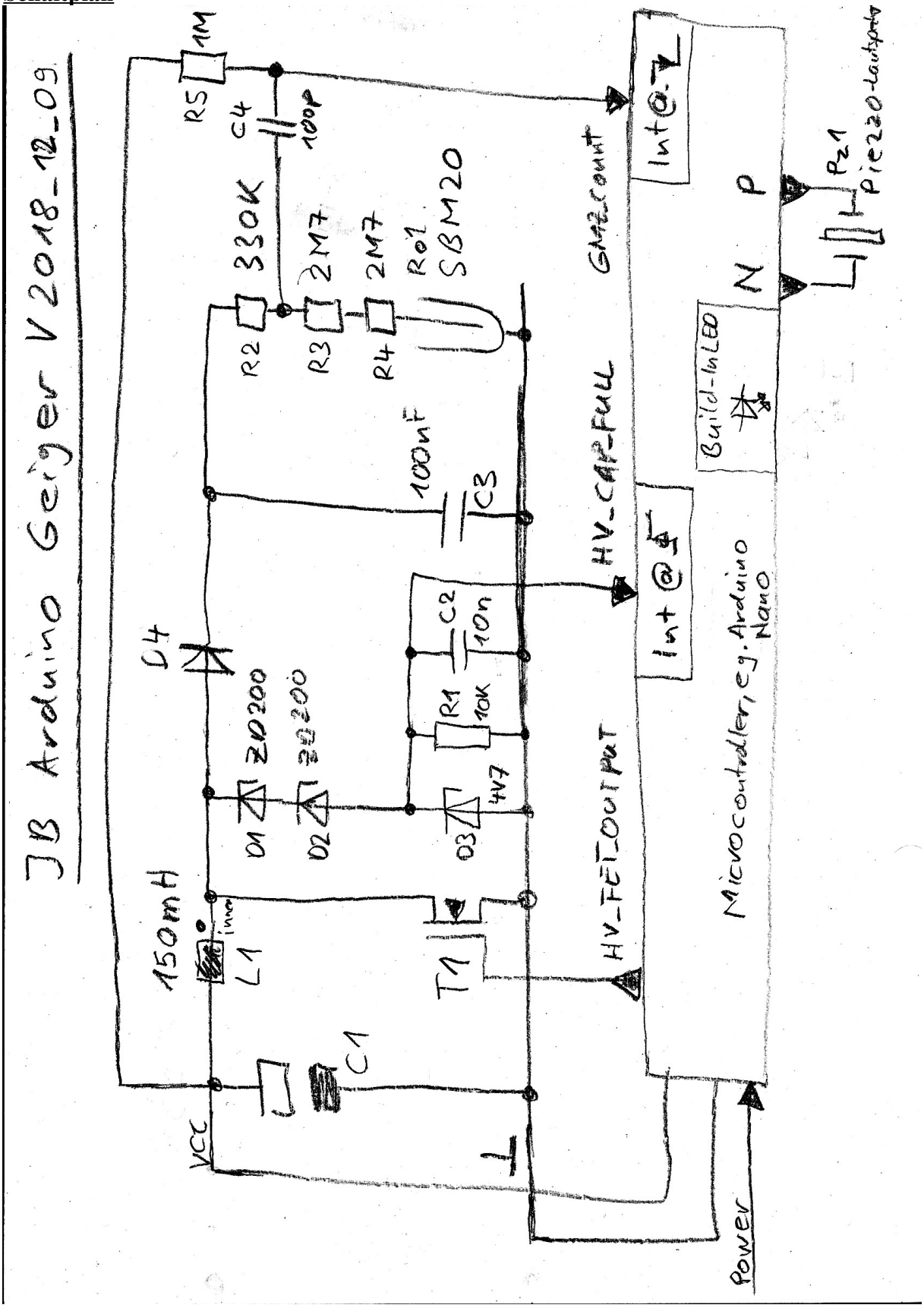
Blockschaltbild

Während ähnliche Konzepte (wie z. B. der „IsiGeiger“, www.opengeiger.de/IsiGeigerDoku.pdf) sehr viel Funktionalität in Form von Schaltungstechnik aufbauen, realisiert der „Simple-Arduino-Geiger“ einen großen Teil auf dem Arduino bzw. in Software. Dadurch wird Software zwar etwas aufwendiger, die Schaltung jedoch wesentlich vereinfacht und ist damit leichter aufzubauen.



Blockschaltbild, rot umrandet: im Arduino realisiert

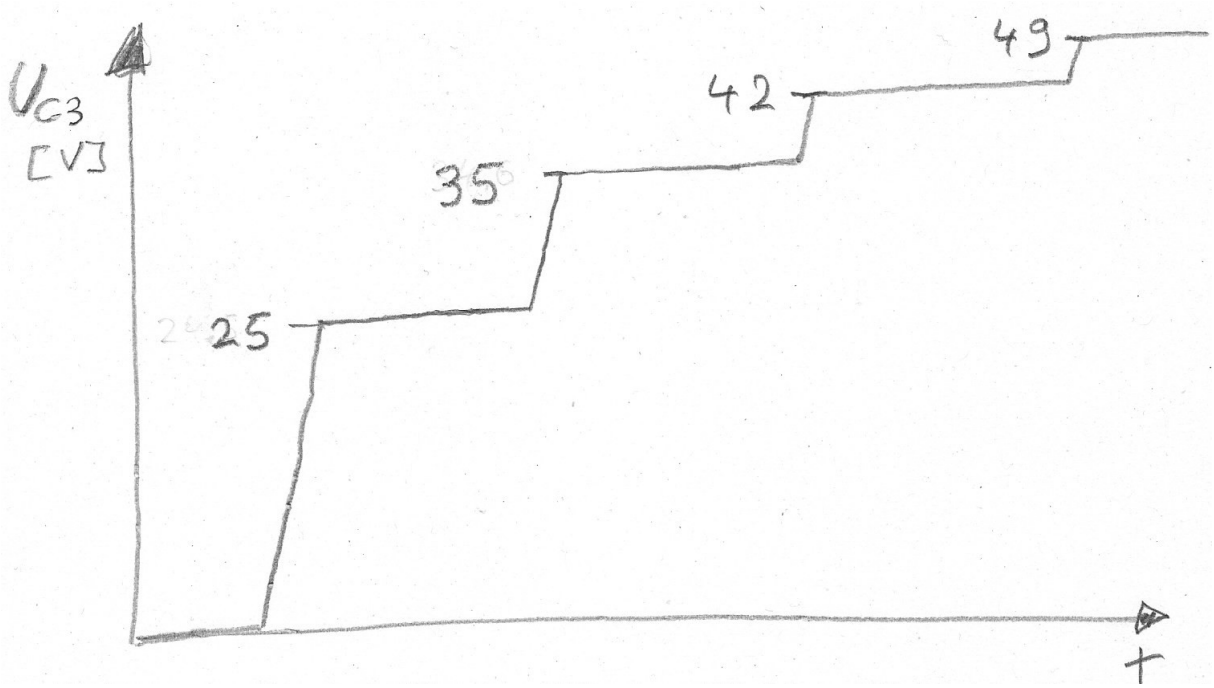
Schaltplan


JB Arduino Geiger V2018_12_09
150mH
L1
C1
VCC
D1
ZD200
D2
ZD200
D3
4V7
L4
T1
R1
10K
C2
10n
R2
330K
R3
2M7
R4
2M7
R5
1M
C4
100p
100nF
C3
R01
SBM20
HV_FET_OUTPUT
HV_CAP_FULL
GM2Count
Int@7
Int@4
Build-InLED
Microcontroller, eg. Arduino Nano
N
P
R21
Piezo-Lautsprecher
POWER

Erzeugung der Hochspannung

Achtung! Auch wenn die im Kondensator C3 gespeicherte Ladung relativ gering ist, sollte man dennoch vorsichtig sein. Die Ladung hält dazu noch einige Zeit an. Will man ganz sicher sein dann sollte das Netz „Plus-Pol“ von C3 unzugänglich gemacht werden. Alle anderen Netze sollten unkritisch sein.

Das Zählrohr Ro1 benötigt eine Spannung von 400 V. Diese wird aus der Betriebsspannung (2,7 ... 5,5 V) mit Hilfe eines Aufwärtswandlers (Boost Converter) erzeugt. C1 dient als Block-Kondensator. Der Mikrocontroller schaltet über den Pin „HV_FET_OUTPUT“ den Feldeffekt-Transistor T1 ein. Dadurch liegt die Spule L1 an Betriebsspannung. Der Strom durch die Spule beginnt zu steigen. Nach einer gewissen Zeit schaltet der Mikrocontroller T1 wieder ab. Allerdings hat sich die Spule L1 an den, durch sie fließenden, Strom derart gewöhnt, dass sie diesen Stromfluss unbedingt weiter aufrecht erhalten will. Um dies zu erreichen erzeugt sie selbst eine Spannung. Somit kann der Strom einen kurzen Moment weiter fliesen. Die Spule ist in der Lage sehr hohe Spannungen zu erzeugen. Der Pluspol der erzeugten Spannung ist der mit dem Punkt markierte Anschluss der Spule. Der aus der Spule kommende Strom fließt über D3 und C3. C3 wird dadurch geladen. Dieser Vorgang wird mehrmals wiederholt. Nach jeder Wiederholung ist die Spannung an C3 etwas höher. Allerdings fällt auf: Der erste Spannungssprung ist sehr hoch. Alle folgenden Spannungssprünge werden immer kleiner.



Die Spannung am Kondensator C3 bei den ersten paar Pulsen

Dies hat folgenden Grund: Wird die Spule bestromt so enthält sie immer die gleiche Energiemenge - die Spulen-Feldenergie:

$$W = \frac{1}{2} * L * I^2$$

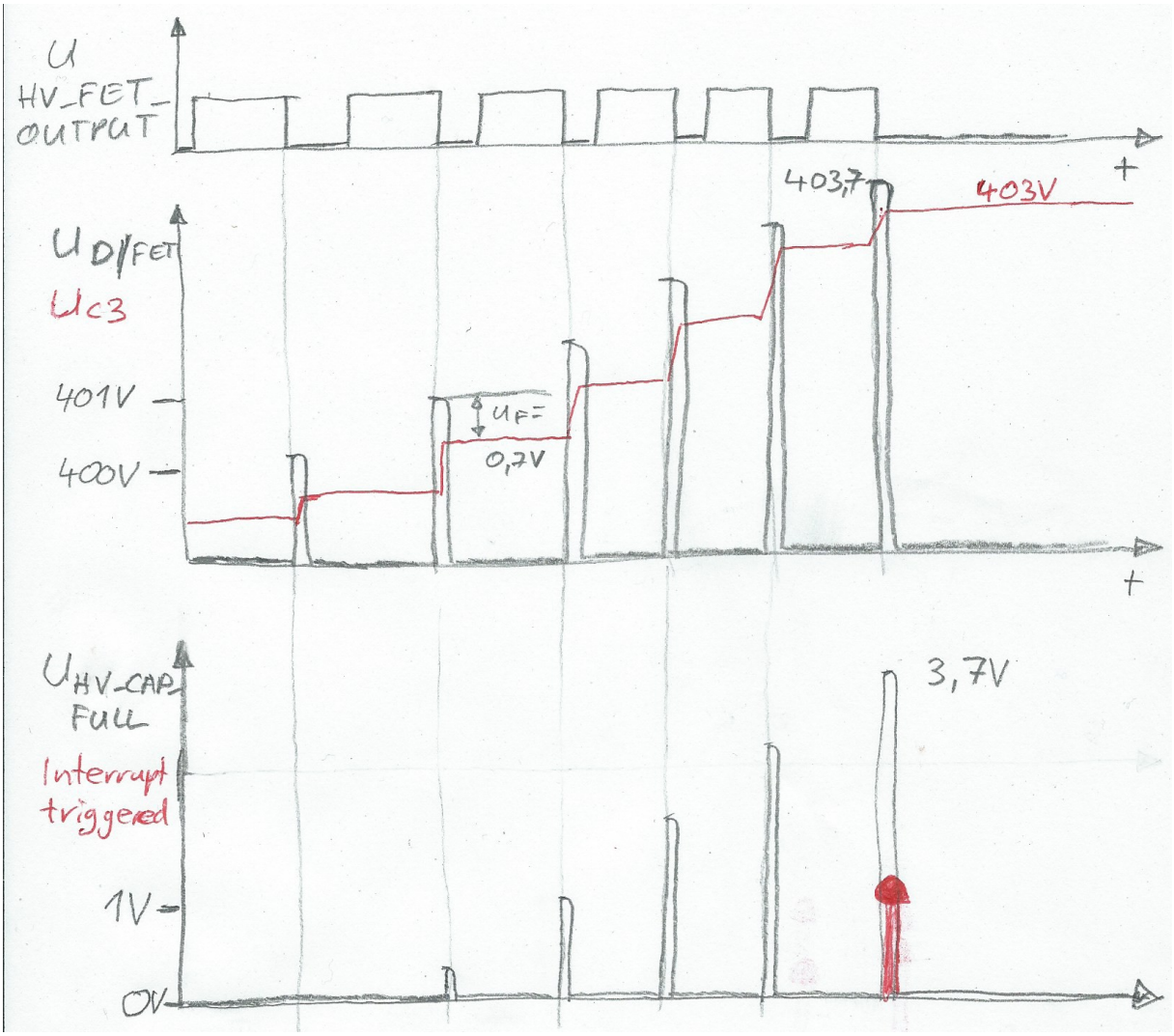
Angenommen der Strom erreicht eine Stärke von 20 mA, so haben wir also eine Energiemenge von 30 uWs =30 uJ (Micro-Joule) in der Spule. Diese Energie wird, idealerweise, komplett an den Kondensator C3 übertragen. Im Kondensator berechnet sich die Feldenergie wie folgt:

$$W = \frac{1}{2} * C * U^2$$

Die erste "Energie-Portion" lässt die Spannung auf ~25 V steigen. Denn bei dieser Spannung beträgt die Energiemenge im Kondensator 30 uJ. Die zweite Energie-Portion erhöht die Energiemenge im Kondensator auf 30 +30 = 60 uJ. Dies entspricht jedoch nicht der doppelten Spannung, sondern (wegen dem quadratischen Zusammenhang) lediglich ~35 V. Und so weiter. So kommt es dass die letzte "Energie-Portion" bei ca. 400 V die Spannung lediglich noch um 0,7 V steigen lässt.

Beendung des Ladevorgangs

Es ist das Ziel den Ladevorgang bei ca. 403 V zu beenden. Hierzu könnte man die Spannung an C3 über einen Spannungsteiler herunter-teilen und diese kleinere Spannung dann weiter verarbeiten. Der Spannungsteiler würde aber ständig C3 entladen und somit sinnlos Energie "verheizen". Daher wird eine andere Strategie gewählt. Am Drain des FETs (U D FET) entstehen lediglich bei den Ladepulsen kurze hohe Spannungspulse. Da D4 ab 0,7 V (Vorwärtsspannung) leitend wird, haben diese Pulse maximal die Höhe von der Spannung an C3 plus 0,7 V. Werden diese Pulse höher als 400 V, so leiten D1 und D2 und die verbleibende Restspannung fällt an R1 ab und ist somit am Pin "HV_CAP_FULL" für den Controller sichtbar. Wird hier, vom Controller, lediglich kurz eine logische "1" erkannt, so wird ein "Interrupt on Change"- Interrupt ausgeführt, der die Variable "GMZ_cap_full" auf "1" setzt und somit den Ladevorgang beendet.



Beendung des Ladevorgangs: Die letzten paar Pulse vor dem Abschalten

Für den Fall dass der Ladevorgang nicht beendet wird, was aufgrund eines Fehlers (Hardware, Programmierung) leicht passieren kann, gibt es die Z-Diode D3. Sie leitet ab 404 V (an C3) den kompletten Strom nach Masse ab und begrenzt somit die Spannung.

Da D1 und D2 keine idealen Z-Dioden sind, haben sie auch eine parasitäre Kapazität. Diese würde dazu führen dass schon bereits weit vor erreichen der gewollten Spannung auf den Eingang "HV_CAP_FULL" triggernde Spannungspulse gekoppelt würden. Dies wird mit C2 verhindert (Kapazitiver Spannungsteiler).

Das Netz am Drain des FETs (U D FET) erzeugt mit seinen Steilen Flanken starke elektrische Felder, die zum einen Funkstörungen verursachen können, die aber auch die eigene Schaltung selbst stören können. Daher sollte dieses Netz im Layout mechanisch so klein wie möglich gehalten werden.

Detektion der Geiger-Zähler-Pulse

Die Schaltung ist so konzipiert, dass die Kathode des Zählrohrs Ro1 auf Masse liegt. Dies hat den Vorteil dass das Netz mit den Pulsen sehr klein gehalten werden kann und somit weniger anfälliger gegen Störungen ist (EMV).

R5 sorgt dafür, dass die Spannung am Pin "GMZ_COUNT" im Normalfall "1" ist. Die Anodenspannung von Ro1 wird über R2, R3, R4 an die Anode gelegt. Empfängt das Zählrohr ein radioaktives Beta-Teilchen oder ein Gamma-Quant, so wird es über die Lawinenentladung nahezu komplett leitend. Dadurch sinkt die Spannung auch im Netz an R2 und R3. Dieser negative Spannungs-Puls wird über den hochspannungsfesten Kondensator C4 an den Pin "GMZ_COUNT" weitergegeben. Nach unten hin begrenzt wird er über die Schutzdioden des Controllers. Der Strom durch die Schutzdioden wird über die Widerstände R2, R3, R4 begrenzt. Am Pin "GMZ_COUNT" wird durch den Puls ein weiterer "Interrupt on Change"-Interrupt ausgeführt, der die Variable " isr_GMZ_counts" hochzählt und die aktuelle "Systemzeit" in die Variable "isr_count_timestamp" schreibt.

Weitere Schaltungs-Teile

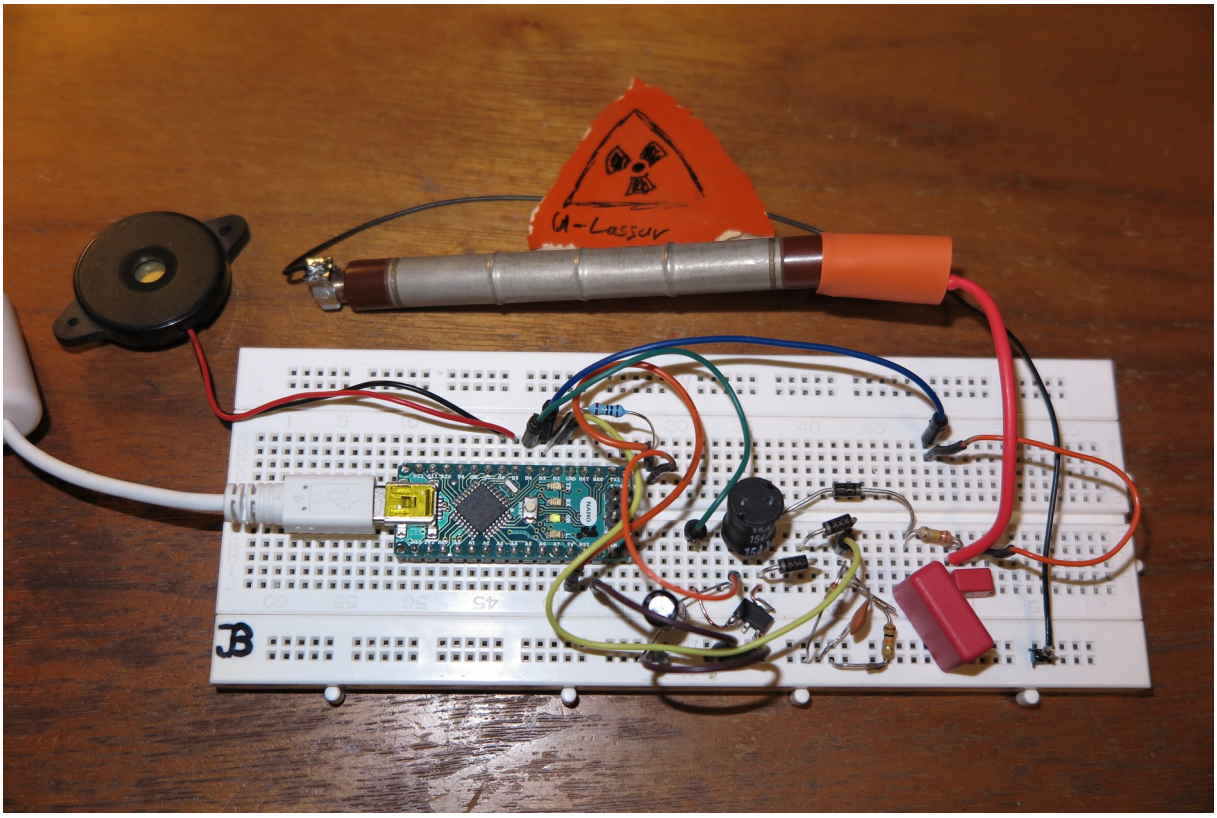
Ist der Parameter "speaker_tick" auf "true" programmiert, so erzeugt der Piezo-Lautsprecher das charakteristische Getacker. Um die Spannung an ihm zu erhöhen wird er nicht gegen Masse betrieben, sondern an zwei Ausgängen, die immer entgegengesetzt geschaltet werden (Gegentakt-Endtufe).

Ebenso wird mit "speaker_tick" auf "true" wird auf der eingebauten LED bei jedem Puls ein Lichtblitz ausgegeben.

Aufbau der Schaltung

Auf Steckplatine

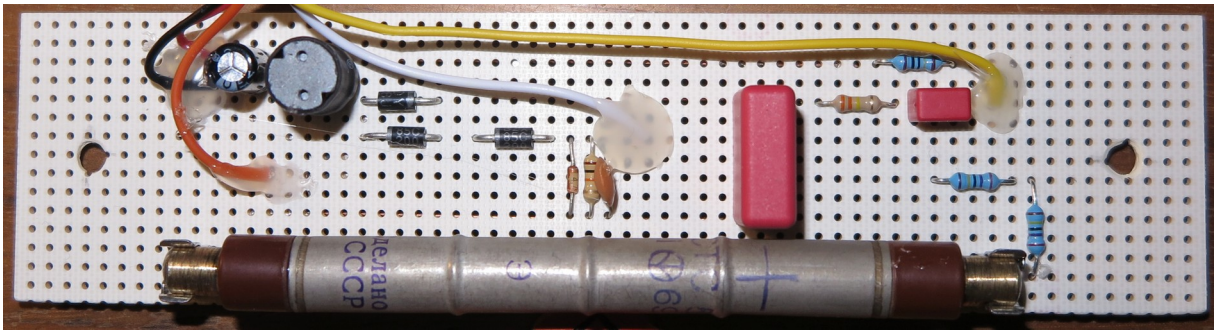
Die Schaltung kann recht einfach auf einer Steckplatine aufgebaut werden. Allerdings muss hierzu der Transistor und die Sicherungshalter (SBM-20-Sockel) angelötet werden. R3 und R4 sind direkt an der Anode angebracht um die Kapazität so klein wie möglich zu halten (erhöht die Lebensdauer des Zählrohrs).



Simple-Arduino-Geiger auf einer Steckplatine (Breadboard)

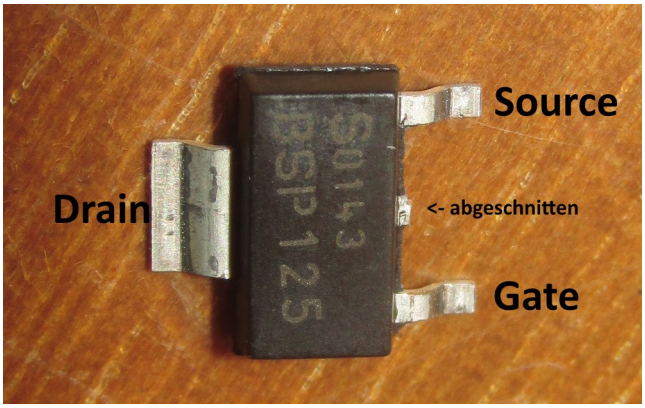
Auf Punkt-Streifenrasterplatine

Wer die Schaltung langfristig verwenden will sollte sie allerdings auf einer Punkt-Streifenrasterplatine aufbauen. Das Netz am Drain des Transistors sollte so klein wie möglich gehalten werden (da es extrem abstrahlt). Daher habe ich den mittleren Drain-Anschluss abgezwickt so dass es nur noch den großen Drain-Anschluss gegenüber von Drain und Source gibt.



Simple-Arduino-Geiger auf einer Punkt-Streifenrasterplatine, Bauteilseite


Simple-Arduino-Geiger auf einer Punkt-Streifenrasterplatine, Lötseite



Der FET-Transistor T1 (BSP125) kann einfach aufgelötet werden. Der Mittlere Anschluss (zusätzlicher Drain-Anschluss) wird abgezwickt.

Simple-Arduino-Geiger auf einer Punkt-Streifenrasterplatine, Layout von Bauteilseite aus

- schwarz: Bauteilseite
- rot: Lötseite

Software

Der „Pin Change Interrupt“ ist eine Hardware-Funktionalität, die auch bei nur sehr kurzen Pulsen an einem speziellen Pin, den Mikrocontroller in eine Interrupt-Serviceroutine schickt. Diese Funktionalität wird sowohl zur Erkennung der kurzen Pulse an PIN_HV_CAP_FULL_INPUT und an PIN_GMZ_count_INPUT verwendet. Die Interrupt-Serviceroutine ist sehr übersichtlich gehalten und setzt, im Falle von PIN_HV_CAP_FULL_INPUT lediglich die globale (volatile) Variable GMZ_cap_full.

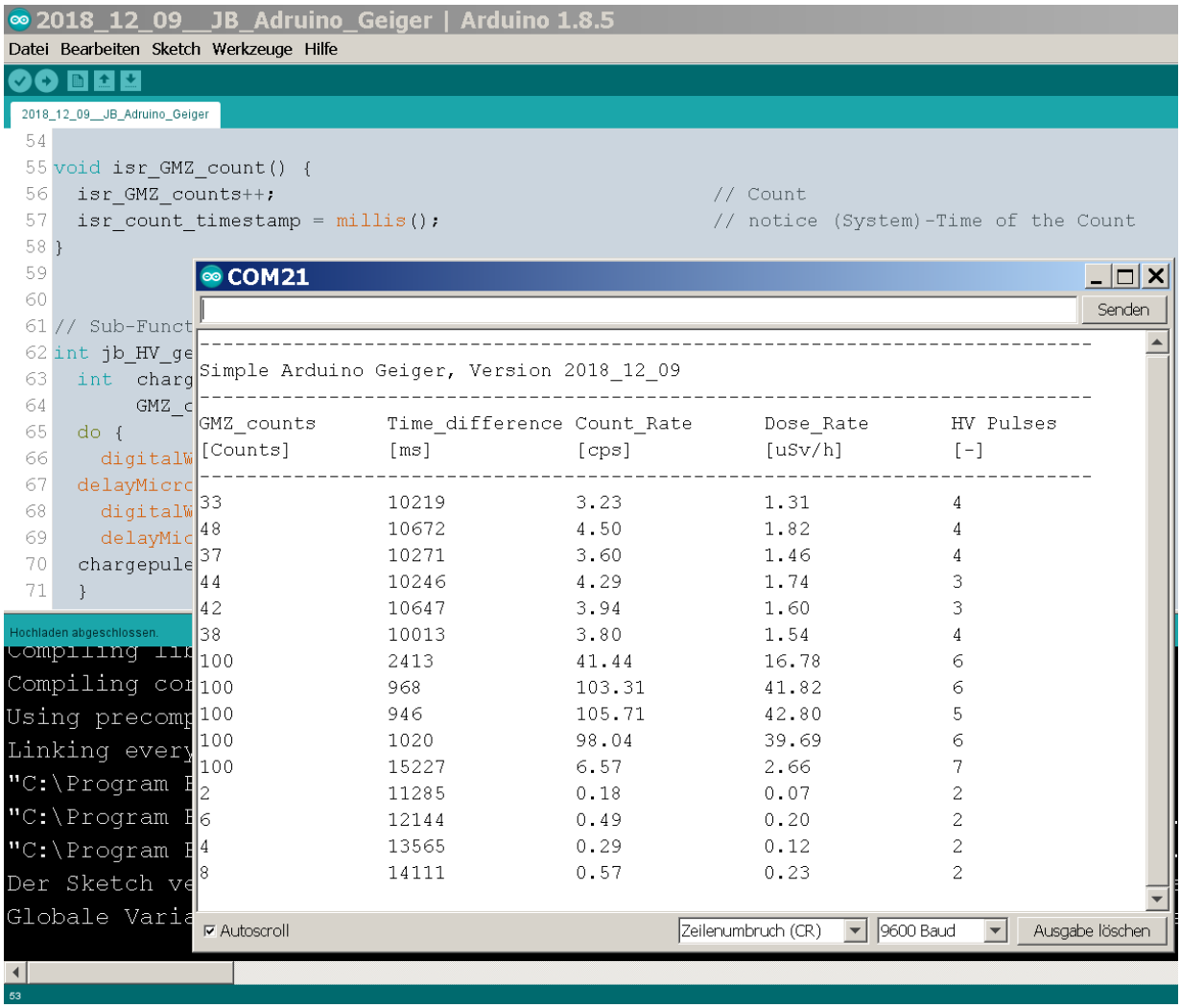
Der Rest der Software erklärt sich weitestgehend von selbst.

Es gibt folgende Parameter mit der Funktionalität ein/ausgeschaltet werden kann:

- debug gibt an ob Werte an die (virtuelle) Serielle Schnittstelle übertragen werden sollen
- speaker_tick gibt an, ob Geigerzähler-Geticke und LED-Geblitze soll ausgegeben werden soll

Die Software (*.ino-Datei) finden Sie auf www.boehri.de. Sie kann weitgehend verändert werden.

Es sollte allerdings darauf geachtet werden dass der Kondensator regelmäßig nach-geladen wird. Ansonsten kann es zu Messfehlern kommen.



Ausgabe auf die (virtuelle) Serielle Schnittstelle, Messung bei drei verschiedenen Strahlungsintensitäten

Stückliste

Die Stückliste finden Sie auf www.boehri.de.

Beschreibungs-Video

Eine Vorstellung der Schaltung sehen sie (vielleicht bald) auf Youtube. Suchbegriffe: Simple-Arduino-Geiger, Jürgen Böhringer