



Středoškolská technika 2016

Setkání a prezentace prací středoškolských studentů na ČVUT

Měření radioaktivity snímáním pulsů GM trubice prostřednictvím Raspberry Pi

*(Measurement of radioactivity by counting pulses of GM tube
through Raspberry Pi)*

Albert Möhwald

Gymnázium Christiana Dopplera
Zborovská 45, Praha

PROHLÁŠENÍ

Prohlašuji, že jsem svou práci vypracoval samostatně a použil pouze podklady uvedené v seznamu literatury vloženém v práci.

Prohlašuji, že tištěná verze a elektronická verze práce jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V dne

Podpis

ANOTACE

Tato práce řeší sestavení detektoru radioaktivity s použitím Geiger-Müllerovy trubice a jeho propojení s Raspberry Pi jako měřič radioaktivity.

Teoretická část zde uvádí všechny teoretické informace související s radioaktivitou a Raspberry Pi.

Praktická část popisuje dvě možná sestavení detektoru radioaktivity, tvarovací obvod pro přenesení impulsů do Raspberry Pi a program pro zaznamenání a zápis impulsů. Nakonec je tu uvedeno i vlastní měření různých předmětů považovaných za možné zářiče.

Práce může posloužit jako návod pro sestavení detektoru radioaktivity nebo jako projekt pro semináře na středních školách.

Klíčová slova: Raspberry Pi, GM trubice, NE555, ionizující záření, Python

ANOTATION

This work deals with radioactivity detector construction using Geiger-Müller tube and its connection with Raspberry Pi as radioactivity counter.

In theoretical part, there is description of all theoretical information related to radioactivity and Raspberry Pi.

Practical part shows two constructions of radioactivity detector, shaping circuit which transfers impulse to Raspberry Pi and Python program for detection and record of measured impulses. There is also an own measurement of various items considered as possible emitters.

This work can be also used as radioactivity detector assembly instructions or as high school workshop project.

Keywords: Raspberry Pi, GM tube, NE555, ionizing radiation, Python

PODĚKOVÁNÍ

Rád bych poděkoval především Petru Möhwaldovi, mému tátovi, za konzultaci a praktické rady k sestavení detektoru radioaktivity. Děkuji také panu profesoru Josefu Pavlovi za užitečné náměty k práci a za poskytnutí dalších prostředků, které mi pomohly s touto prací.

OBSAH

Úvod.....	6
1 Teoretická část.....	7
1.1 Radioaktivita	7
1.1.1 Ionizující záření	7
1.1.2 Jednotky radioaktivity	8
1.1.3 Dávka.....	8
1.2 Detektor radioaktivity.....	10
1.2.1 Ionizace.....	10
1.2.2 Geiger-Müllerova trubice	10
1.3 Raspberry Pi	12
1.3.1 Konektory GPIO.....	12
1.4 Linux	14
1.5 Python.....	16
2 Metodika	19
3 Praktická část	21
3.1 Obvod pro trubici STS-5	21
3.2 Testovací obvod pro blikání LED diodou	22
3.3 Testovací obvod pro vstup tlačítka.....	24
3.4 Tvarovací obvod s NE555	26
3.5 Zapojení Raspberry Pi s detektorem	27
3.6 Program měření a zpracování dat	27
3.7 Výpis dat.....	31
3.8 Výsledky vlastního měření radioaktivity vybraných předmětů	33
3.9 Výsledky vlastního měření radioaktivity v určitých lokalitách ČR	35
Závěr a diskuze.....	36
Literatura	37
Přílohy	39
Datasheet trubice STS-5	44

ÚVOD

Práce je průřezem různých oborů informatiky a fyziky. V informatice lze toto téma zařadit do aplikované informatiky, ve fyzice patří do dozimetrie a elektroniky. Práce mne velmi motivovala k získání základních znalostí o operačním systému Linux a využití malého počítače Raspberry Pi k propojení s detektorem radioaktivity, jež obojí mám doma. Tímto zařízením mohu s menší nepřesností určit, jakou radioaktivní dávku poskytuje jaký materiál nebo prostředí, a to s možností grafických znázornění. Bez počítače by bylo pouze možno říci, zda je daná látka nebo prostředí radioaktivní nebo ne.

V teoretické části vysvětluji základní informace od fyzikálních veličin přes zařízení až po fungování programovacích jazyků a operačních systémů, které všechny využiji v práci. V praktické části spojím všechny tyto souvislosti dohromady, vysvětlím zapojení Raspberry Pi s detektorem, popíši programy, algoritmy a nakonec zobrazím data měření.

KAPITOLA 1

TEORETICKÁ ČÁST

Nejprve je nutno podat základní informace vztahující se k tématu měření radioaktivity a Raspberry Pi do souvislostí. Tato kapitola stručně pojednává o radioaktivitě (druhy ionizujícího záření, jednotky radioaktivity v souvislosti s účinky na člověka, měření radioaktivity) a o Raspberry Pi včetně jeho využitého softwaru (Linux, Python) a periférií (konektor GPIO).

1.1 Radioaktivita

Radioaktivita je schopnost látky, při které dochází k přeměně atomových jader nebo ke ztrátě energie atomového jádra. Dochází k uvolňování ionizovaných částic a paprsků.

Radioaktivita byla objevena náhodou v roce 1895 francouzským fyzikem **Antione Henri Becquerel** při výzkumu fosforescence (dlouhodobé luminiscence). Fosforescence nastávala pouze tehdy, pokud byl materiál osvětlen, a tím byly atomy vybuzeny do excitovaného stavu, což se projevovalo vyzařováním materiálu. Nicméně při studiu uranové soli Becquerel zjistil, že sůl vydává „neviditelné“ záření i bez předchozího nasvícení. Uran tedy zářil z jiného, vlastního zdroje energie. [1]

1.1.1 Ionizující záření

Ionizujícím zářením rozumíme vysokoenergetické záření, schopné vyrazit elektrony z atomového obalu, a tak způsobit ionizaci látky (viz 1.2.1. Ionizace).

Hlavní fyzikální veličinou charakterizující ionizující záření je jeho **energie**. Ionizující záření nezáří pouze na jedné energii, ale ve více energiích, jež jsou charakterizovány spektrem záření, stejně jako o ostatních elektromagnetických záření, pomocí frekvence nebo vlnové délky. Pomocí planckovy konstanty můžeme jednoduše vyjádřit z energie frekvenci, a tak pomocí analýzy energie záření vynést jeho spektrum. Pokud má záření **spojité spektrum**, jedná se o záření beta, pokud má **čárové spektrum**, je to gama záření. Energie tohoto záření je udávána v *elektronvoltech*. Převodní vztah mezi *jouly* a *elektronvolty* je:

$$1 \text{ eV} = 1,602 \cdot 10^{-19} \text{ J}$$

Existuje několik druhů záření, podle pronikavosti a chování v elektrickém a magnetickém prostředí:

Záření alfa je svazek letících jader atomů helia. Alfa záření patří do skupiny přímo ionizujících záření (záření nesoucí elektrický náboj, jehož kvanta přímo odtrhávají elektrony z

atomového obalu) má ve vzduchu krátký dosah a je velmi slabě pronikavé. K zastavení záření postačí list papíru. S účinky záření alfa je také podobné méně časté **protonové záření** s jedním letícím protonem.

Záření beta se skládá z letících elektronů (β^-) nebo pozitronů (β^+). Toto záření – zařazováno také mezi přímo ionizující – je až stokrát pronikavější [1], než záření alfa. Beta částice zastavílobal, plíšek olova, lepící páska, oblečení, ...

Záření gama jsou vysoce energetické elektromagnetické paprsky a je jedním z nejpronikavějších záření. Záření je nepřímio ionizující (záření předávající svou energii elektricky nabitě částici, jež následně ionizují látku). K odstínění záření se obvykle využívá tlustá vrstva olova, či beton.

Důležitým zářením reagujícím přímo s atomovými jádry je **neutronové záření** vznikající při jaderných explozích nebo uměle v jaderných reaktorech. [2] Z hvězd a výbuchů supernov vzniká **neutrinové záření**, záření téměř nereagující a zásadní pro astronomii. Tato záření jsou mnohem pronikavější než gama záření.

Nejčastějším přírodním zdrojem záření je **uran 238**. Uran má velmi dlouhý poločas rozpadu (4,5 miliardy let). Jadernými reakcemi uranu vznikají další prvky – thorium, polonium, radon – vzácný radioaktivní plyn, jenž je nebezpečný při výskytu v ovzduší, kvůli jeho usazování v plicích, způsobující rakovinu plic. [3] Dalšími zdroji ionizujícího záření jsou **elektronické zářiče** (urychlovače částic, rentgenové trubice), **zářiče z kosmu** (hvězdy, explodující supernovy, ...), **elektrické výboje** (blesk). [4]

1.1.2 Jednotky radioaktivity

Základní jednotkou radioaktivity s soustavě SI je *becquerel* [Bq]. Tato jednotka vyjadřuje **aktivitu zářiče** (zn. A) a je definována jako počet jader přeměněných za jednotku času, nebo ekvivalentně úbytek počtu jader dosud nepřeměněných. [1]

Protože jeden Bq je velmi malou jednotkou existuje starší, větší a známá jednotka *curie* [Ci], (podle manželů M. S. Curie a P. Curie). Převodní vztah mezi Ci a Bq je:

$$1 \text{ Ci} \cong 3,7 \cdot 10^{10} \text{ Bq}$$

1.1.3 Dávka

Dávkou je v tomto smyslu myšlena **pohlčená energie** udávaná veličinou **absorbovaná dávka** (zn. D). Jednotkou absorbované dávky SI soustavy je *gray* [Gy] (podle anglického radiologa R. H. Graye). Tato jednotka vyjadřuje, kolik energie ionizujícího záření bylo absorbováno jednotkou hmotnosti. 1 Gy je ekvivalentní 1 J/kg.

V souvislosti s dozimetrií (obor zabývající se účinky záření) je dávka měřena ve vztahu s účinky na lidskou tkáň. Číselné určení absorbované dávky je složité, protože závisí na druhu ionizujícího záření a na látce pohlcující záření. V dozimetrii je zavedena tzv. **ekvivalentní dávka** (zn. H), jednotka používaná dnes a pravidelně při určení míry záření. Ekvivalentní dávka je součinem absorbované dávky, faktorů záření a faktorů modifikujících. [5] Nejpoužívanější a zároveň jednotkou SI je *sievert* [Sv] (podle švédského fyzika R. M. Sieverta), tedy J/kg. V USA je více než Sv používána starší jednotka *rem* (**r**öntgen **e**quivalent **i**n **m**an). Převod mezi Sv a rem je:

$$1 \text{ Sv} \cong 100 \text{ rem}$$

Pro lepší představu jsou v tabulkách níže vypsány hodnoty dávek:

0,05 μSv	Dávka při spaní s někým ¹
0,1 μSv	Dávka banánu
5- 10 μSv	Jedna sada rentgenových snímků zubů
30 mSv	Jedna prohlídka těla na CT (hlava - 2 mSv, hrud' - 7mSv)
68 mSv	Odhadovaná max. dávka, kterou dostali evakuovaní lidé žijící v blízkosti elektrárny Fukušima
670 mSv	Maximální dávka, jež dostali pracovníci řešící havárii ve Fukušimě
6 Sv	Smrtelná dávka při krádeži radioaktivního zářiče pro radioterapii v brazilském městě Golanía
21Sv	Smrtelná dávka, jež dostal Louis Slotin nehodou při experimentování na atomové zbrani
64Sv	Zatím největší známá dávka, jež kvůli špatné diagnóze dostal Albert Stevens v průběhu 21 let vpíchnutím experimentální dávky plutonia při Projektu Manhattan [8]

TABULKA 1: PŘÍKLADY DÁVEK [3], [6]

< 1 $\mu\text{Sv/h}$	Bezpečné; normální úroveň dávky
2 $\mu\text{Sv/h}$	Velmi nízké riziko; dávka v letadle ve výšce do 10 km
5 $\mu\text{Sv/h}$	Nízké riziko; dlouhodobý pobyt v oblasti nebezpečný
20 $\mu\text{Sv/h}$	Vysoké riziko; nebezpečí choroby
1 mSv/h	Vysoké riziko; nutná okamžitá evakuace
100 mSv/h	Velmi vysoké riziko; otrávení radiací
> 10 Sv/h	Smrtící; selhávání orgánů, smrt v do několika hodin

TABULKA 2: ZDRAVOTNÍ RIZIKA PROSTŘEDÍ [7]

¹ Dávka při spaní s někým je ekvivalentní dávka, již získá dospělý člověk od jiného dospělého člověka, kde se oba lidé nachází v jejich nejkratší vzdálenosti od sebe menší než 50 cm po dobu osmi hodin. Dávku obdrží oba lidé. Dávku, již získají více než dva lidé splňující podmínky výše, se nezabývám.

1.2 Detektor radioaktivity

1.2.1 Ionizace

K detekci radioaktivního záření je zapotřebí využít vlastnosti **ionizace**. Ionizace je děj, při kterém vzniká elektricky nabitý ion z elektricky neutrálních atomů nebo molekul. Ionizaci způsobuje např. kosmické záření, radioaktivní záření, atd. kdy se ionizovaná vysoce energetická částice srazí s neutrální částicí. Ionizovaná částice dodá svou energii proti ionizační energii (energie potřebná dodat k uvolnění jednoho elektronu) neutrálního atomu/molekuly. Pokud je energie ionizované částice vyšší, nastane ionizace. Prvky vzácných plynů mají největší ionizační energii, a tak jsou nejobtížněji ionizovatelné. [9] Viz obrázek 1.

1.2.2 Geiger-Müllerova trubice

Geiger-Müllerova trubice obsahuje anodu (připojena na +), katodu (připojena na kostru), a je vyplněna elektricky vodivým plynem, vhodným k ionizaci. K umožnění impulzu uvnitř trubice je napětí mezi anodou a katodou je vysoké (okolo 400 – 1000V).

Částice přilétající do trubice způsobí ionizaci vnitřního plynu. Ionizovaná molekula/atom vypustí elektron, jenž získá energii z elektrostatického pole a způsobí další ionizaci. Tímto způsobem vzniká tzv. elektronová lavina schopná svým vzniklým výbojem přímo propojit anodu s katodou. [10]

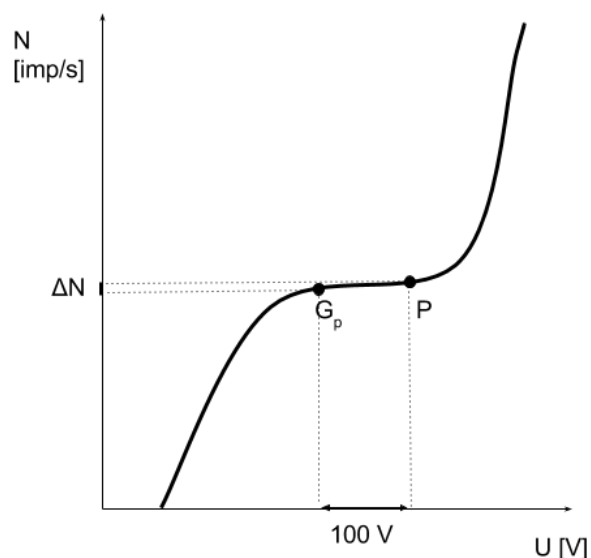
Pokud je měřena radioaktivita zářiče, v radioaktivním materiálu probíhá A rozpadů za sekundu. Tuto hodnotu nelze v mikrofyzičce považovat za absolutně přesnou. Veličina udává pouze nejpravděpodobnější průměrný počet rozpadů za sekundu. Měření je tedy zatíženo **odchylkou**, kvůli níž je počet částic dopadajících na GM trubici udáván ve tvaru počtu dopadů s odchylkou její odmocniny a to celé za čas.

$$N = \frac{n}{t} \pm \frac{\sqrt{n}}{t}$$

Přesnost měření také určují tři základní parametry GM trubice:

- **Charakteristika**
 - Je to závislost počtu pulsů na napětí na GM trubici.
 - Na grafu je křivka s plošinou.
 - Pokud je napětí pod Geiger-Müllerovým prahem G_p , pak jsou impulsy závislé ještě i na energii ionizující částice, a tak ne každé kvantum může vyvolat dostatečný lavinový výboj.

- Je-li napětí mezi body G_p a P, pak jsou zaznamenány všechny impulsy uvnitř trubice. Interval těchto bodů se nazývá **plošina** a je charakterizována **strmostí** v %/V a **délkou** V. K měření je napětí na detektorech vždy v tomto intervalu.
- Při napětí nad bodem P dochází k nedostatečnému zhášení lavinových výbojů, a tak dochází k poškození trubice a snižování její účinnosti.



- **Mrtvá doba**

- Udává dobu, do kdy po zaznamenání impulsu nemůže být zaznamenán další impuls.

- **Účinnost**

- Udává poměr zaregistrovaných částic C ku poměru částic dopadajících na plochu trubice. Pokud budeme uvažovat vyzařování zářiče na všechny strany, tedy do tvaru koule, můžeme pak spočítat geometrický faktor jako poměr plochy zakřiveného obdélníku (vypočítaný z části plochy kulové úseče) a povrchu koule. Součin aktivity A a geometrického faktoru udává výsledný teoretický počet částic dopadající na trubici. Výsledný výpočet účinnosti je následující:

$$\mu = \frac{C}{A \frac{\Omega}{4\pi r^2}}$$

1.3 Raspberry Pi

Raspberry Pi (dále RasPi) je počítač velikosti mobilního telefonu sestavený plošnými spoji na jedné destičce. Obrovskou výhodou je jeho nízká cena (100 - 1300 Kč). RasPi byl vyvinut v roce 2012 britskou společností Raspberry Pi Foundation za účelem podpořit zájem studentů o informatiku a ukázat funkci řízení jiných zařízení (web: <https://www.raspberrypi.org>). Operačním systémem je Linux (Raspbian, Arch, Ubuntu, ...). Nicméně RasPi může fungovat i s operačním systémem Windows, a to Windows 10 IOT Core. Pro vzhled Raspberry viz obrázek 2.

Dnes již existuje více druhů RasPi:

Raspberry Pi 1 (původní RasPi) má procesor BCM2835 se 700 MHz, paměť (SDRAM) 256 MiB; grafický procesor Video Core IV s podporou OpenGL ES 2.0, 1080p30 a MPEG-4; výstupy HDMI, TRS, 3,5mm jack, RCA; 1 USB 2.0 port; slot na microUSB kartu periférie 26 GPIO pinů, SPI, I²C, I²S, UART; a je napájen microUSB při 300mA. RasPi 1 má modely A, A+, B, B+. Výhodou modelu B je jeho možnost na síťové připojení ethernetem a větší počet USB portů (model B má 2 porty, B+ má 4 porty). Modely A+, B+ mají navíc 40 GPIO pinů. Jeho dnešní cena pro model B+ je přibližně 800 Kč.

Raspberry Pi 2 (uveden v únoru 2015) je podstatně výkonnější, než RasPi 1. Obsahuje stejný nebo lepší hardware, a tak poskytuje i více softwarových možností. Procesor má 900MHz s paměť 1024 MiB. Stejně jako RasPi model B+ má 4 USB porty a 40 GPIO pinů. K dispozici je více operačních systémů včetně varianty Windows 10 IoT Core. Dnes stojí přibližně 1300 Kč.

Raspberry Pi Zero (uveden v listopadu 2015) je menší než původní řada RasPi. Z hardwaru má pouze jedno-jádrový 1GHz procesor s 512 MB RAM (o 40 % rychlejší než RasPi 1), HDMI, microUSB port, microUSB napájení a 40 GPIO pinů. RasPi Zero zatím není na českém trhu. Dnes (přelom února a března 2016) je kvůli jeho nízké ceně (4 £ nebo 5 \$) na britských a amerických e-shopech zcela vyprodán. [12]

Raspberry Pi 3 (uveden v únoru 2016) je třetí generací nahrazující RasPi 1 B+ a RasPi 2. Na rozdíl od RasPi 2 má 64bitový 1,2 GHz procesor ARMv8 a bezdrátové síť 802.11n LAN, Bluetooth 4.1 a Bluetooth Low Energy. [13]

Samozřejmě pro využití nedostupných funkcí (Wi-Fi, VGA, Bluetooth, ...) lze na RasPi připojit i další desky. Neoficiálně vznikly rozšiřující desky např. Getboard, UniPi board, ... [14]

1.3.1 Konektory GPIO

GPIO (*General purpose input/output*) jsou piny, jež lze ovládat pomocí softwaru RasPi. U RasPi jsou vývody různých typů: dva 5V0 (napájení 5 V), dva 3V3 (napájení 3,3 V), pět GND

(zem), GPIO (samotné vývody ovládané RasPi). Navíc je RasPi opatřeno univerzálním přijímačem/vysílačem UART (*Universal Asynchronous Receiver/Transmitter*), rozhraním SPI (*Serial Peripheral Interface*) a sběrnici I²C (*Inter-Integrated Circuit*). Tyto komunikace lze překonfigurovat na 9 určitých GPIO vývodů ze 17 GPIO vývodů celkem. Viz obrázek 3.

Pomocí programu **LXTerminal** nebo programovacích jazyků Python, C++ (potřeba stáhnout knihovnu Wiring Pi) nebo dalších je možné piny přepínat do stavů LOW a HIGH, tedy na logickou 0 a 1. Pokud není GPIO pod napětím 0 V nebo má napětí až 0,7 V, napětí je vyhodnoceno jako úroveň 0 (LOW). Napětí od 2,7 V do 3,3 V je vyhodnoceno jako úroveň 1 (HIGH). Přivedení napětí vyšší jak 3,3 V do vstupního pinu způsobí okamžité zničení RasPi. Interval od 0,7 V do 2,7 V je hazardním stavem, kdy GPIO může být vyhodnoceno jako jeden ze stavů LOW i HIGH.

Proudová zatížitelnost pinů závisí na modelu RasPi. Model A má spotřebu 500 mA, a tak pin 5V0 dodá z rozdílu napájení při maximálně 1A proud 500 mA. Spotřeba modelu B je 700 mA, pin 5V0 dodá tedy nejvýše 300 mA. Pin 3V3 může u všech RasPi dodávat maximálně proud 50 mA. Do GPIO pinů může vtékat a vytékat proud do 16 mA. Větší proud zničí připojený pin. [15]

1.4 Linux

Linux je operační systém vytvořený ze systému UNIX v roce 1991 finským programátorem **Linusem Torvaldsem** a skupinou programátorů po celém světě. Maskotem Linuxu je sedící tučňák. Název získal Linux spojením jmen Linus a posledního písmena z UNIX.

Dnes již je Linux vysoce nebezpečným konkurentem firmy Microsoft a jejím operačním systémům Windows. Linux má **free ware** licenci a je možné ho využívat zdarma. Navíc je velice stabilní, málokdy dojde k selhání systému. Jeho instalace většinou obsahují základní kancelářské, grafické vývojové a další programy, které Windows neobsahují. Nicméně průběh instalace operačního systému je značně náročnější než instalace Windows.

Jedním z nepříjemných překvapení může být nemožnost spuštění programů s příponou **.exe**. Tyto programy spustí pouze Windows, DOS, OS/2 nebo je lze spustit pomocí programů určených pro unixový kernel (Wine, Cedega). [16] Když není jednoduše možné používat **.exe** soubory, jak lze nainstalovat nový program? Obvykle existují dvě možnosti. První možností je stáhnout a spustit instalační balík. Druhou možností je využít příkazu **apt-get** (*advanced packaging tool*) spuštěním z konzole (fungují na distribucích Debian a Ubuntu, na jiných nemusí fungovat):

```
sudo apt-get install [název programu]

sudo apt-get update           //možné zadat poprvé při instalaci, nebo
sudo apt-get upgrade         //když systém nebyl aktualizován
```

Po zadání příkazu k instalaci systém vyhodnotí soubor a volné místo na disku a zeptá se, zda chcete balík nainstalovat. Potom stačí zadat **Y** a balík bude nainstalován.

Ovládání Linuxu je zcela určitě efektivnější než ovládání především novějších verzí Windows. Zatímco ve Windows se používají pouze okna, myš a někdy klávesnice, v Linuxu se využívá **terminál** (konsola). Terminál se ovládá pomocí krátkých příkazů. Pro začínajícího uživatele bude toto zpočátku náročné, ale po čase, kdy využije základních příkazů, to bude snadnější. Zde je přehled několika základních příkazů:

```
cd [adresa]                   //změní aktuální adresář
ls                             //vypíše soubory a složky v aktuálním adresáři
mkdir [název]                 //vytvoří nový adresář se zadaným názvem
top                             //vypíše probíhající procesy
sudo                           //vykoná příkaz jako super-user
./[program]                   //spustí zkompilovaný program
cat [soubor]                  //vypíše obsah souboru do konzole
nano [soubor]                 //spustí soubor v textovém editoru NANO
tar -zxvf [soubor].tar.gz     //rozbalí komprimovaný soubor
```

Pokud nechce uživatel využívat **cd** a **ls**, může využít programu pro správu souborů, který jistě bude součástí základní instalace. Pokud by s tím nebyl spokojen, může si nainstalovat program

Midnight Commander podobný programu Total Commander spouštějící se z konzole pomocí příkazu *mc*. Příkaz *sudo* se nedoporučuje používat, pokud to není nezbytně nutné. Tento příkaz povoluje provádět změny možné nenávratně poškodit operační systém.

Z hlediska využívání internetu je Linux značně stabilnější než Windows. Servery poskytující internetové stránky většinou používají linuxové systémy. Vždyť Linux je vyvíjen programátory po celém světě, a tak jistě věnovali část svého úsilí do stabilního internetu, kvůli neporuchovému sdílení své práce. Je tedy zřejmé, že podpora, spolehlivost, rychlost budou vynikající, a navíc drtivě překovává Windows. Chyby se mohou vyskytnout v prohlížeči, jenž je součástí instalace (např. Conqueror) a nemá nejnovější internetové protokoly. Řešením je instalace jiného prohlížeče (Mozilla Firefox, Opera, ...).

Proti virům a jiným škodlivým útokům je Linux celkem imunní. Ke změně systémových souborů nebo souborů jiných uživatelů je zapotřebí použít příkaz *sudo*, který z internetu nelze jednoduše použít.

Na rozdíl od Windows má Linux mnoho distribucí vyvinutých velkými i malými společnostmi. Distribuce jsou vytvářeny podle požadavků na druh počítače, na hardware, náročnosti na ovládání nebo uživatelské náročnosti. Nejpoužívanějším dělením distribucí jsou **desktopové** (vytvořené pro uživatele) a **serverové** (vytvořené pro efektivní funkci počítače, který může být nevšední). Nejrozšířenější distribucí je Ubuntu. Další distribuce se nazývají např. Debian, Fedora, OpenSuse, Arch, Slax, Gentoo ...

Hlavní nevýhodou Linuxu může být nepodporování ostatními výrobci softwaru, již vytváří své programy především pro nejpoužívanější celosvětovou platformu Windows. Hry a jiné aplikace nelze bez použití speciálního programu spustit nebo se prostě nespustí. V únoru 2013 byla oficiálně zprovozněna herní platforma Steam na distribuci Ubuntu, umožňující hrát populární hry na Linuxu. K hraní her může být užitečné mít na počítači oba operační systémy.

1.5 Python

Python je open source skriptovacím jazykem vyvinutý v roce 1991 nizozemským programátorem **Guidem van Rossumem**. Jazyk byl vymyšlen již o vánocích roku 1989, kdy měl Rossum dovolenou. Název Python vznikl kvůli autorově oblíbenosti britské komediální skupiny Monty Python.

Jazyk je **multiplatformní**, instalace je možná na většině operačních systémů (včetně Windows, Linux, Mac OS). Většina distribucí Linuxu obsahuje ve své základní instalaci vývojové prostředí IDLE.

Python existuje v propojení s dalšími programovacími jazyky. Kvůli výkonu jazyka C existuje nejpoužívanější interpretace CPython používající se k zdokonalování samotného Pythonu. Nejnovější implementací pro vývoj platforme Microsoft .NET Framework je implementace v podobě jazyka IronPython. [17]

Jazyk C je **kompilovaným** jazykem. Jeho výsledný kód se musí k vykonání přeložit kompilátorem do tzv. strojového kódu, podle kterého procesor počítače vykoná zkompilevané příkazy. Kód daného programovacího jazyka musí být napsán přesně tak, aby jej kompilátor správně analyzoval. Jakoukoli syntaktickou chybu v jediném znaku kódu nedokáže kompilátor zkompilevat. V psaní kódu se může programátor také dopustit logické chyby a nebo chyby vyžadující nevyhrazenou paměť procesoru. Takovéto chyby kompilátor zkompileje, nicméně při průběhu programu může dojít k přetečení proměnných, operování s náhodnými čísly, pádů programu a další. Oproti kompilovaným jazykům existují **interpretované** jazyky, jež nekompilejí zdrojový kód do strojového kódu, ale jednotlivé příkazy kódu se interpretují do strojových příkazů. Proto jsou interpretované jazyky většinou přívětivější pro programátory svou jednodušší syntaxí a snadnějším spuštěním. [18] Python je interpretovaným jazykem používající částečně i kompilaci. Při spuštění programu Python zkompileje kód do efektivnějšího interpretovaného mezikódu probíhající v průběhu programu. Navíc Python po prvním spuštění pouze porovnává směny v kódu a podle toho si překompileje svůj mezikód. Proto tedy lze program mnohem snadněji spustit, má přehledný kód, neuvádí složené závorky, nicméně při velkých operacím může být v řádu milisekund pomalejší. [19]

Zdrojový kód Pythonu je především typický svou jednoduchostí a přehledností. **Odsazování** je zde nezbytné. V jiných jazycích jsou sekvence příkazů uzavřeny do složených závorek (C, Java Script) nebo zahájeny a ukončeny příkazy (Paskal). Kód Pythonu je navržen tak, aby byl k programátorům přívětivý. Namísto závorek nebo klíčových slov využívá přesného odsazení, které jsou jednotné pro jednu sekvenci příkazů. Příkladem je toto větvení:

```
x = input("Napište spam: ")
if x == 'spam':
```

```

print ("Zdravím spame.")
print ("Je příjemné počasí, že ano?")
print ("Přeji krásný den.")
else:
print ("Toto není spam.")
print ("Dnes není váš dobrý den, že ne?")

```

Dalším prvkem jeho jednoduchosti je automatické **definování datového typu** při deklaraci proměnné (jako v PHP). V některých jazycích je nutné před operacemi s proměnnými definovat datový typ dané proměnné. V Pythonu se datová proměnná definuje již při přiřazení čísla nebo textového řetězce. Nicméně při vzájemných operacích s proměnnými někdy může být nutné předem definovat datový typ. Například nelze sčítat typ integer s typem string.

```

x = input('Zadejte x: ')
y = x + 1
print("Hodnota x zvýšena o 1 je:", y)
-----
--
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:16:59) [MSC v.1900 32
bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Admin\Desktop\test.py
=====
Zadejte x: 2
Traceback (most recent call last):
  File "C:\Users\Bertik\Desktop\...py", line 2, in <module>
    y = x + 1
TypeError: Can't convert 'int' object to str implicitly

```

Zde je vidět, že pokud by došlo k pokusu o sčítání typu integer a string, Python vypíše v průběhu programu konkrétní chybu.

Od 3. prosince 2008 byl zveřejněn **Python 3**. Python 2.7 je předešlou verzí, ze které přešel nový Python. Python 2.7 i Python 3.0 jsou a budou v průběhu aktualizovány. Python 3.5.1 je dnes nejaktuálnější verzí Pythonu 3. Verze 2 se především liší od verze 3 změnou příkazu *print* na funkci *print()* a v dalších změnách názvů funkcí. [20]

```

Python 2: print "Odpověď je", 42
Python 3: print("Odpověď je", 42)

Python 2: print x,           # Čárka nutí program k neukončení řádku
Python 3: print(x, end=" ") # Zde se nastaví, co má být na konci řádku

Python 2: x = raw_input(Zadej x: )
Python 3: x = input(Zadej x: )

Python 2: input('Zadej kód: ')
Python 3: eval(input('Zadej kód: '))

```

Operační systém Linux nemusí rozpoznat interpretující jazyk. Kvůli mnoha variacím instalace systému Linux může být interpret jazyků umístěn nestandardně. Navíc programátor vždy nemusí

chtít interpretovat program v jazyku Python, ale v jiném jazyce. Proto je v systému Linux bezpečnější psát na začátek souboru tento komentář:

```
#!/usr/bin/env python3
```

Tento řádek obsahuje adresu s umístěním interpretu. Při spuštění programu v terminálu se spustí jako příkaz:

```
python3 program.py
```

Pokud by byl program spuštěn jiným způsobem a bez komentáře, může dojít ke spuštění jako souboru jako strojového kódu: [21]

```
./program.py
```

V programu vždy nemusí fungovat kódování (např. při psaní v Linuxu). Proto se na začátek kódu dává komentář:

```
# -*- encoding: utf-8 -*-
```

KAPITOLA 2

METODIKA

V této nejkratší kapitole popíší součástky potřebné pro sestavení detektoru a tvarovacího obvodu. Měřič radioaktivity lze rozdělit na tři části:

- **Raspberry Pi** vvhodnocující obvod s možností napájení detektoru.
- **Tvarovací obvod** propojující RasPi s detektorem a vyhlazující impulzy z detektoru.
- **Detektor radioaktivity** (s trubicí STS-5).

Měřicí přístroje:

- Komponenta vlastního měřiče - trubice STS-5
 - Minimální napětí pro detekci = 280...330 V
 - Doporučené pracovní napětí = 360...440 V
 - Délka plošiny = alespoň 80 V
 - Strmost plošiny = 0.125% / 1 V
 - Maximální přírodní pozadí = 27 pulsy / min
- Měřič Wolfcraft GAMMA CHECK - A

Obvod detektoru radioaktivity na 5V napájení:

- GM trubice STS-5
- Transformátor (z tiskárny, primár: 25 a 20 závitů, sekundár: 1200 závitů)
- Potenciometr (500 k Ω)
- PNP Tranzistor (C559B)
- NPN Tranzistor (BC547B)
- NPN Tranzistor (3x BC546B)
- Dioda (2x)
- LED dioda (žlutá)
- Zenerova dioda (2x)
- Rezistor (7,8 M Ω ; 100 k Ω ; 51 Ω)
- Magnetický bzučák

Tvarovací obvod:

- Časovač NE 555 (zapojený astabilně jako tvarovač impulsů)
- Optron (odděluje obvod na část s napájením 5V a 3V3)
- Kondenzátory (10 pF, 100 nF, 22 nF)
- LED diody (červená, zelená)
- Rezistory (187 k Ω , 8,6 k Ω , 560 Ω , 100 Ω)

Měřené zářiče:

- Wolframová elektroda WT30 (s obsahem 3% oxidu thoria)
- Rtuťová výbojka TESLA RVK 250W
- Demonstrační zdroj záření DZZ GAMA 241 Am, 300 kBq, 60 keV
- Váza z uranového skla (zakoupená ve vetešnictví v Praze - Podolí)

Pro vykreslení obvodů byl využit program *Circuit Simulator v1.6d.cz* www.falstad.com

KAPITOLA 3

PRAKTICKÁ ČÁST

Nyní jsem potřeboval využít všechny předchozí uvedené poznatky, zapojit obvod, napsat program pro uložení impulzů, data zanalyzovat a přehledně vypsát.

3.1 Obvod pro trubici STS-5

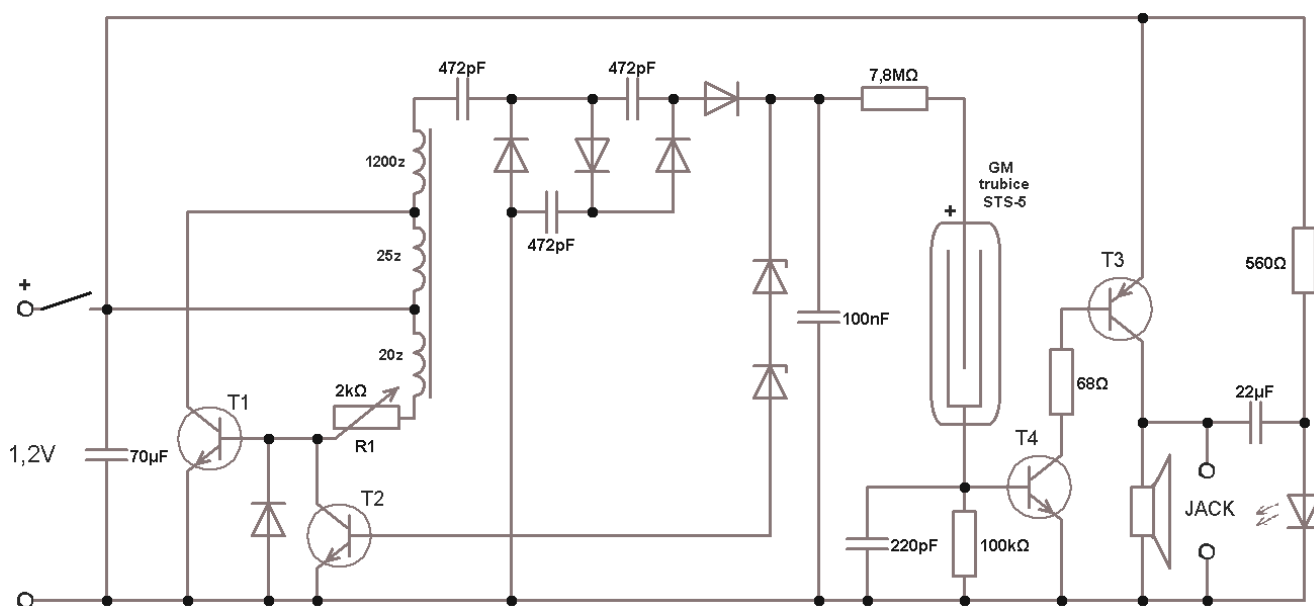
Hlavní součástí obvodu je Geiger-Müllerova (GM) trubice **STS-5** (rusky CTC-5), vyrobená v bývalém Sovětském svazu, využívána v detektorech radioaktivního záření (např. se využívá v československém měřiči RBGT-62A). Dnes není trubice v České republice téměř dostupná, mnou byla koupena přibližně za 420 Kč přes internetovou aukci eBay z Ukrajiny. Trubice detekuje záření beta a gama při napětí 350 - 400V. Datasheet (v ruském jazyce) je k dispozici v příloze, charakteristika je v metodice.

Další důležitou součástí je **transformátor**, vyjmutý ze staré kancelářské tiskárny fungující jako oscilující násobič napětí s velmi zesilujícím tranzistorem TR1 kladných půlvln. K řízení oscilace je využíván potenciometr R1.

Nízká spotřeba detektoru je výhodou. Odběr proudu je 20 mA a může být správným sestavením násobiče snížen až na 4 mA.

Výstup detektoru je opatřen konektorem JACK 3,5mm přenášejícím impuls do sluchátek nebo do tvarovacího obvodu. Impulzy GM trubice jsou detekovány sluchátkem (magnetickým bzučákem) a červenou svítivou LED diodou.

Tento obvod je sestaven podle publikace na stránkách elektrikáře a domácího kutila Dana z Mělníka. [11]



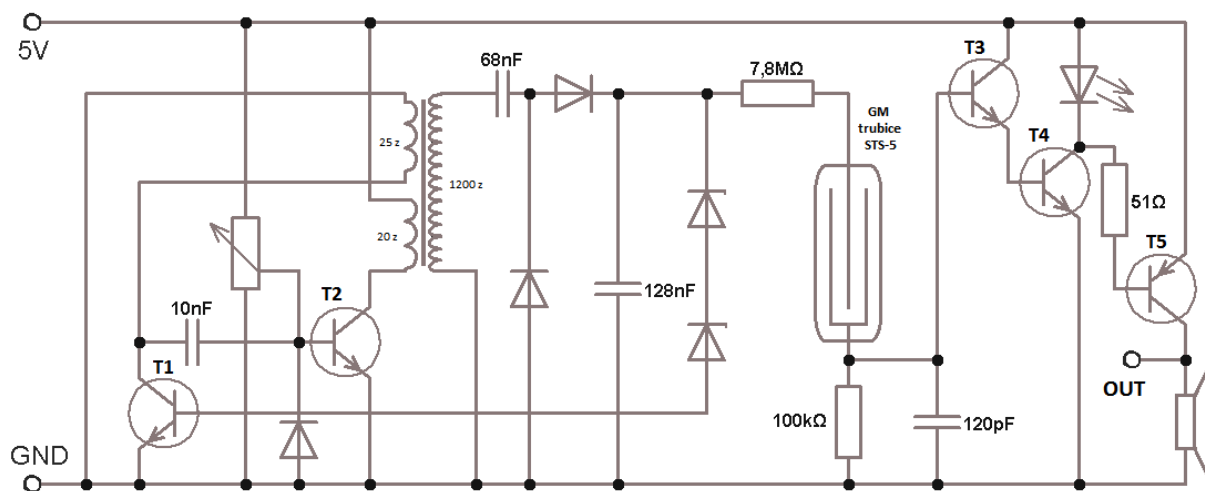
Po zapnutí detektoru jsou již vydávány impulsy bez zářiče. Detektor indikuje zbytkové **kosmické záření** a může indikovat i radioaktivní prvky vyskytující se ve vzduchu nebo v půdě (izotopy radonu, cesia, draslíku, ad.). Podle ruského datasheetu by nemělo **pozadí** překročit 27 pulsů na minutu.

Touto hodnotou je možné určit **kalibrační konstantu** umožňující s nepřesností odhadnout aktuální pohlcenou ekvivalentní dávku. Za rok obdrží každý v ČR průměrně 1,5 mSv/h, což je přibližně 0,17 $\mu\text{Sv/h}$. Podíl těchto čísel nám udává kalibrační konstantu:

$$\frac{0,17 \mu\text{Sv/h}}{27} \doteq 0,0063$$

Jelikož k počítání pulsů používám Raspberry Pi, je výhodné napájet detektor radioaktivity napětím 5 V přímo z RasPi; uvádím zde zapojení pro napájení 5 V.

Tento obvod funguje téměř stejně, jako obvod výše. Potenciometr (500 k Ω) je nastaven tak, aby byl proud, při pracovním bodu tranzistoru T2, na kolektoru 10 mA. Namísto násobiče napětí předchozího obvodu nyní postačil zdvojovač napětí. Svítivost LED diody a zvuk bzučáku je zesílen Darlingtonovým zapojením, jejíž tranzistor T3 a následně tranzistory T4 a T5 jsou otevřeny impulsem v GM trubici.



3.2 Testovací obvod pro blikání LED diodou

Nejprve bylo potřeba zajistit ovládání výstupů RasPi. První zkouškou možnosti ovládání jsem zvolil řízené rozsvícení a zhasnutí LED diody. K tomu mi stačilo použít rezistor, LED diodu, kontaktní pole a napsat softwarový program na spínání diody.

Zelenou LED diodu s pracovním úbytkem napětí 2,2V zapojím mezi výstupem GPIO a svorkou GND. Výstup GPIO v úrovni HIGH poskytuje napětí 3,3 V. Při připojení LED diody přímo na vývod by došlo k definitivnímu poškození RasPi vysokým proudem. Proto je nutno do

série připojit rezistor, jenž bude mít odpor takový, aby obvodem procházel proud dostatečný na rozsvícení LED diody (2 mA) a nepřesahující maximální rozsah proudu, jenž může vytékat vývodem GPIO (do 16 mA). Rozhodl jsem se pro bezpečný proud 2 mA. Níže je jednoduché schéma zapojení a výpočet potřebného elektrického odporu:

$$U_{LED} = 2,2 V$$

$$U_{GPIO} = 3,3 V$$

$$I = 2 mA$$

$$R = ?$$

Kirhoffův zákon o uzavřené smyčce: $U_{GPIO} = U_{LED} + U_R$ [1]

$$\text{Ohmův zákon: } R = \frac{U_R}{I} \text{ [2]}$$

Dosazením [1] do [2] vznikne:

$$R = \frac{U_{GPIO} U_{LED}}{I}$$

$$R = \frac{3,3 V - 2,2 V}{0,002 A} = 550 \Omega$$

Po výpočtu vychází odpor na 550 Ω. Nejbližší vyšší rezistor v odporové řadě je rezistor s odporem 560 Ω, používám tedy tento.

Aby bylo možné napsat program, musel jsem si stáhnout ovladače GPIO na Python. Tehdy byly k dispozici na:

<https://code.google.com/p/raspberry-gpio-python/downloads/list>, nicméně dnes tam již nejsou.

Řešením by snad mohly být příkazy v konzole LXTerminal:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install python-dev python-rpi.gpio
```

Nyní už mi stačilo napsat zdrojový kód v Pythonu. K ovládání GPIO pinů je nutné nahrát modul *RPi.GPIO*. Zásadní je o tomto modulu vědět: nezbytnost určení číslování pinů, nastavení GPIO vývodu na vstupní nebo výstupní režim a příkazy pro zapnutí a vypnutí úrovně LOW nebo HIGH u výstupního vývodu. Základ kódu pro řízení GPIO je takový:

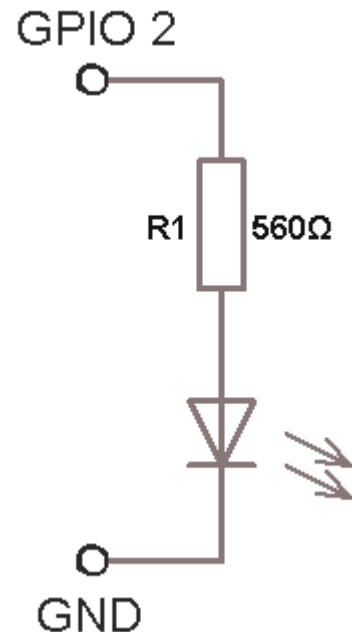
```
#!/usr/bin/env python3
# -*- encoding: utf-8 -*-

import RPi.GPIO as GPIO #nahraje modul pro ovládání vývodů

GPIO.setmode (GPIO.BCM) #nastavení označení GPIO jako v procesoru
GPIO.setwarnings (False)

#Kód programu

GPIO.cleanup() #vrátí vývody do původních stavů
```



Příkaz nastavující vývod GPIO na výstupní nebo vstupní režim je:

```
GPIO.setup([pin], GPIO.OUT)
GPIO.setup([pin], GPIO.IN)
```

Po nastavení GPIO vývodů již je možné zapínat a vypínat jednotlivé vývody:

```
GPIO.output([pin], GPIO.HIGH)
GPIO.output([pin], GPIO.LOW)
```

Nakonec bylo potřeba využít funkce *time.sleep([proměnná])* z modulu *time*, jež zastaví průběh kódu v místě zavolání na určený čas. Celkový kód se vstupem nastavení času svícení diody a výstupem svícení je:

```
#!/usr/bin/python3
# -*- encoding: utf-8 -*-

import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(4, GPIO.OUT)

x = float(input('Zadejte kolik sekund má LED svítit: '))
GPIO.output(4, GPIO.HIGH)
print("Rozsvícení LED")
time.sleep(x)
GPIO.output(4, GPIO.LOW)
print("Zhasnutí LED")

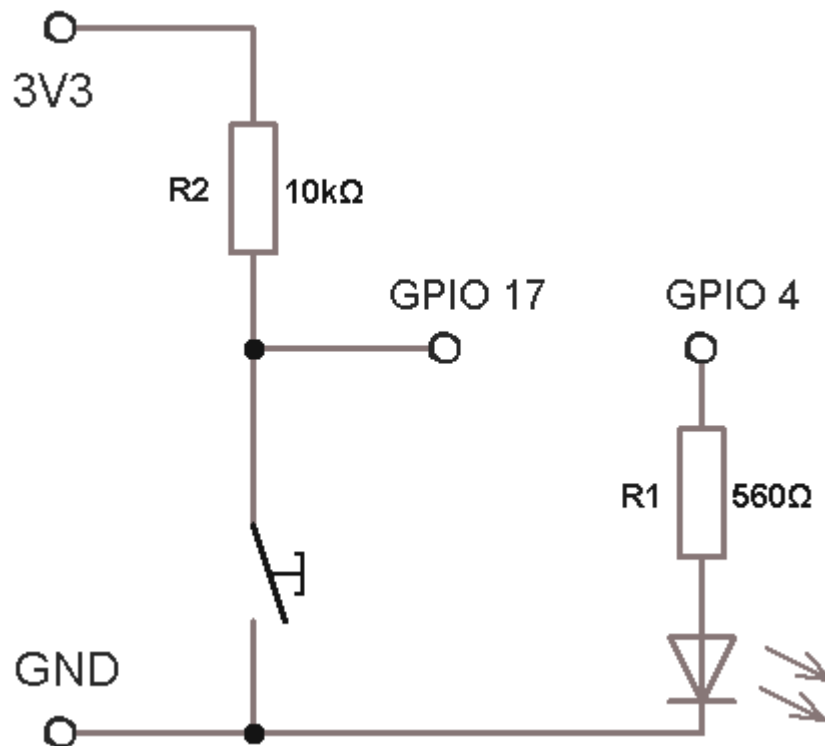
GPIO.cleanup()
```

[22]

3.3 Testovací obvod pro vstup tlačítka

V předchozím obvodu jsem změnami logických úrovní LOW a HIGH výstupu GPIO vývodu ovládal LED diodu. K následujícímu jsem využil obvodu předchozího a přidal jsem tlačítko, kterým spínám LED diodu. K tomu bylo zapotřebí další GPIO vývod, tentokrát nastavený na vstupní režim.

Vstup má logickou hodnotu 1, pokud na něm je napětí 2,7 V až 3,3 V. Stisknutím tlačítka nabude logické 0, protože zdroj RasPi bude propojen na kostru, obvodem bude procházet proud a napětí téměř 3,3 V bude na rezistoru R2. Toto lze přirovnat k situaci člověka dotýkajícího se elektrického ohradníku a k ptákoví sedícímu na elektrickém drátu bez kontaktu se zemí.



V obvodu samozřejmě musí být rezistor R2. Pokud by nebyl v obvodu, došlo by k přímému zkratu a RasPi by bylo trvale zničeno. R2 může mít teoreticky minimální odpor 206,25 Ω. Z bezpečnostních důvodů RasPi jsem raději připojil odpor 10 kΩ, abych měl jistotu, že na odporu bude vysoké napětí, aby opravdu došlo k přepnutí logické úrovně. [23]

Kód pro řízení tlačítkem obsahuje část předchozího kódu, modul *time*, cyklus neustále spouštějící svou sekvenci příkazů dokud aktuální čas nenabude zadaný čas a větvení. Podmínkou rozsvícení LED diody je *GPIO.input(17) == False*. Zde je nutné si uvědomit, že po stisknutí tlačítka je napětí na GPIO 17 téměř nulové, proto hodnota *False*.

```
#!/usr/bin/python3
# -*- encoding: utf-8 -*-

import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.OUT)
GPIO.setup(17, GPIO.IN)

x = int(input('Zadejte dobu snímání vstupu v sekundách: '))
timeout = time.time() + x
while (time.time() < timeout):
    if(GPIO.input(17) == False):
        GPIO.output(4, GPIO.HIGH)
    else:
        GPIO.output(4, GPIO.LOW)

GPIO.cleanup()
```

Výsledně tento program v určeném čase kontroluje, zda je velmi malé napětí na vývodu GPIO 17 (je stisknuté tlačítko). Pokud je tato podmínka splněna, program rozsvítí LED diodu. V opačném případě LED diodu ponechá vypnutou.

V programu je také využil funkce `time.time()` z modulu `time`. Zavolání této funkce vrací aktuální čas v sekundách v datovém typu `float`. Pokud nastavím proměnnou `timeout` (využívám výše ve zdrojovém kódu) na `time.time() + x`, kde `x` byl požadovaný čas měřerní, mohu tak kvůli neustále aktualizující se funkci, jejíž hodnota stoupne až na `timeout`, načasovat podmínku cyklu `while`, jehož sekvence bude neustále zkoušet, zda je na vývodu GPIO 17 napětí.

3.4 Tvarovací obvod s NE555

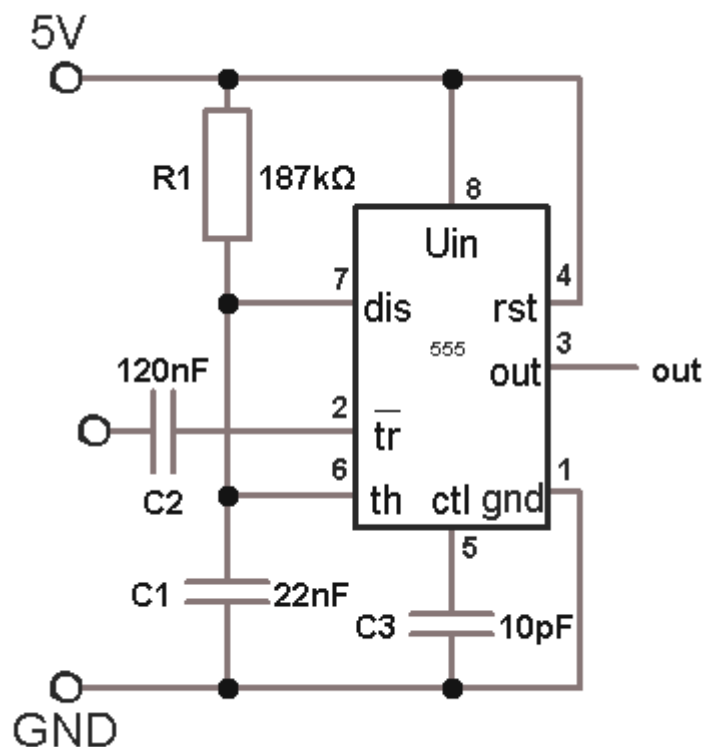
Když jsem měl vyzkoušený výstup i vstup RasPi, zbývalo mi propojit detektor s RasPi. Cílem bylo spočítat impulsy vytvořené detektorem radioaktivity v určitém čase a vyjádřit z nich počet impulsů za minutu. Jednoduchým řešením by bylo propojit výstup detektoru mezi vývodem (např. z předtím používaného GPIO 17) a kostrou. Impulsy detektoru nemusí být silné, přesné a vyhlazené. Jeden impuls by mohl být zaznamenán jako více impulsů.

K vyhlazení impulsů jsem využil časovače NE555 (viz obrázek 6). 555 je integrovaný obvod skládající se ze dvou komparátorů a jednoho klopného obvodu. Primárním využitím této součástky je přesné časování výstupu signálu. Vedlejším

využitím může být zapojení jako tvarovač obdélníků, převodník, měřič frekvencí, ... Já jsem NE555 využil pro stabilizační účely jako tvarovač neurčitých impulsů detektoru do přesných obdélníkových impulsů. Schéma je podle monostabilního zapojení sestavené podle free ware softwaru 555 Timer (1997) napsaného Adrewem Clarkconem (viz obrázek 8 v příloze). Doba signálu na výstupu se vypočítá jako součin odporu na R1 a kapacitě kondenzátoru na C1:

$$t = \ln(3) CR$$

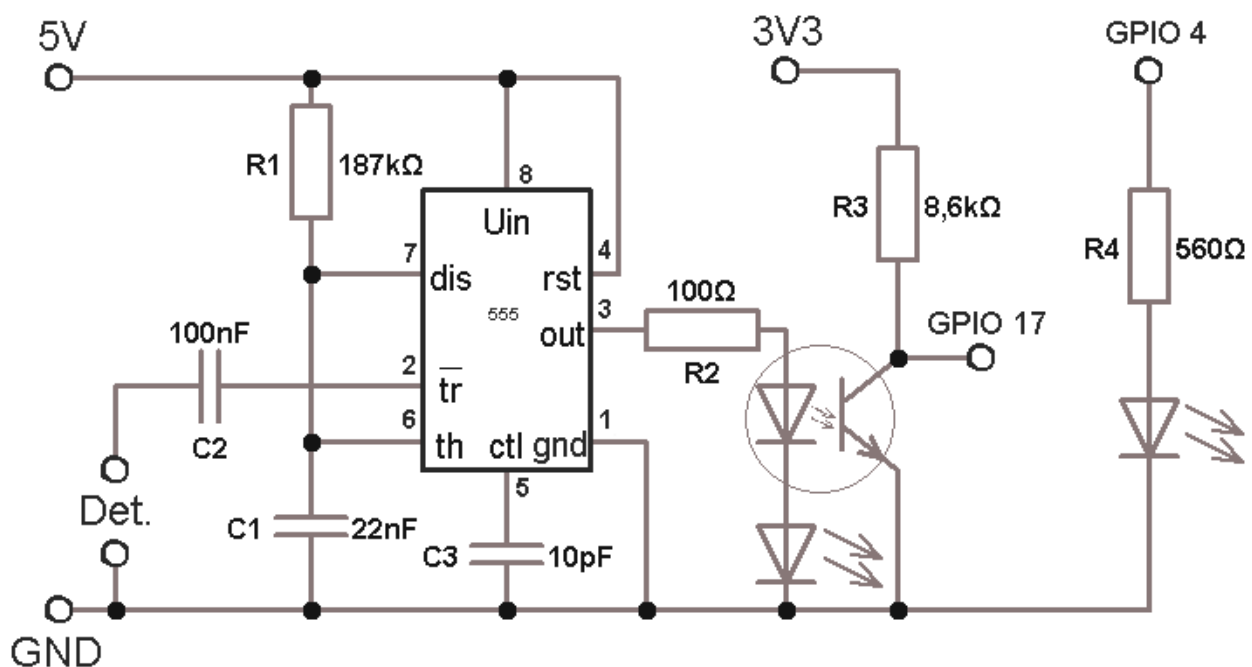
Pro tento obvod vychází čas 4,5 ms. Tato doba by neměla být kratší, než jeden cyklus zaznamenávajícího programu. Tento malý čas znamená, že RasPi může zaznamenat maximálně okolo 220 impulsů za sekundu. [24]



3.5 Zapojení Raspberry Pi s detektorem

V tuto chvíli jsem měl připraveno připojení tvarovacího obvodu a obvodu pro vstup detektoru. Na kontaktním poli bylo ještě potřeba zajistit správné napájení. Časovač 555 je napájen při napětí od 4,5 V, kdežto na RasPi v obvodu pro vstup se nesmí přivést napětí vyšší než 3,3 V (vyšší napětí by dostalo RasPi do hazardní úrovně).

Tvarovací obvod a obvod pro vstup detektoru je potřeba oddělit. K tomuto účelu jsem využil optronu, uzavřené fotoelektrické součástky skládající se z LED diody a NPN fototranzistoru. Pokud je dioda optronu napájena, vyše optický paprsek a otevře fototranzistor. Pokud tedy detektor vyše impulz, 555 vyhledá impulz a ten pak pokračuje do diody optronu. V okamžiku, kdy diodou optronu projde signál, otevře se fototranzistor a druhou částí obvodu začne procházet proud z napájecího vývodu 3,3 V do kostry, dokud se na výstupu 555 neobjeví 0. V tomto



obvodu tedy optron nahrazuje tlačítko testovacího obvodu pro vstup.

Ze schématu je vidět, že dochází ke spojení tvarovacího obvodu s obvodem pro vstup spínačem. Změna tu nastala pouze v nahrazení tlačítka optronem a přidáním zatěžujícího odporu R2. Do obvodu je možné přidat LED diodu indikující signál 555 (viz obrázku 8 v příloze zapojení).

3.6 Program měření a zpracování dat

Možností, jak napsat program bylo více. Zvolil jsem programovací jazyk Python kvůli jeho jednoduchému modulu a podpoře zápisu do souborů. Mohl jsem zvolit jazyk C, jenž je rychlejší, nicméně modul *RPI.GPIO* je snazší k naprogramování než knihovna *WIRING.PI*.

Na vstupu jsem měl měnící se logické hodnoty *true* a *false* na pinu GPIO 17. Tyto hodnoty se měnily v určitém čase, takže při tom bylo potřeba zaznamenat čas měření. K vyhodnocení dávky, tedy zatím k vyhodnocení počtu impulsů za jednotku času, bylo samozřejmě potřeba sčítat jednotlivé impulsy a mít program časově omezen. Dávka, kterou obdrží trubice, se vypočítá jako celkový počet pulsů za celkový čas měření.

```
#!/usr/bin/python3
# -*- encoding: utf-8 -*-

import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.OUT)
GPIO.setup(17, GPIO.IN)

x = float(input('Zadejte dobu měření v minutách: '))
timeout = time.time() + x*60
impulz = int(0)
while (time.time() < timeout):
    if(GPIO.input(17) == False):
        GPIO.output(4, GPIO.HIGH)
        impulz = impulz + 1
        print("Impulz: ", impulz)
    else:
        GPIO.output(4, GPIO.LOW)
radiace = impulz/(x*60)
print("Celkem", impulz, "impulzu za", x, "min.")
print("Radiace:", radiace, "pulsu/s")

GPIO.cleanup()
```

Tento kód, napsaný výše, by naměřil mnohem více impulsů, než kolik by vyslal detektor. Proč? V momentu zaznamenávání impulsu stihne procesor RasPi vykonat několik desítek cyklů zaznamenávající jednotlivé impulsy. Proto je potřeba zavést proměnnou, která se při prvním cyklu záznamu impulsu změní na hodnotu 1 a po ukončení příjmu signálu se navrátí na 0. Tuto proměnnou jsem nazval níže *zaznamenano*.

Do následujícího zdrojového kódu jsem přidal výpis naměřené dávky při každém impulsu v průběhu měření. Zavedl jsem tři proměnné. První proměnnou *predcas*, do níž jsem uložil čas na začátku měření. Druhá proměnná *aktrad* vyjadřuje aktuální stav měření s použitím okamžitého počtu impulsů a okamžitého času získaný rozdílem aktuálního času a proměnné *predcas*. Poslední proměnná *cas* vyjadřuje okamžitý čas měření rozdílem aktuálního času a proměnnou *predcas*.

```
#!/usr/bin/python3
# -*- encoding: utf-8 -*-

import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
```

```

GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.OUT)
GPIO.setup(17, GPIO.IN)

x = float(input('Zadejte dobu měření v minutách: '))
timeout = time.time() + x*60
predcas = time.time()
impulz = int(0)
zaznamenano = 0
while (time.time() < timeout):
    if(GPIO.input(17) == False):
        GPIO.output(4, GPIO.HIGH)
        if(zaznamenano == 0):
            impulz = impulz + 1
            aktradm = round(impulz/(time.time()-predcas), 2)
            cas = round(time.time()-predcas, 1)
            print("Impulz:", impulz, "Radiace:", aktradm, "Cas:", cas, "/",
x*60)
                zaznamenano = 1
        else:
            GPIO.output(4, GPIO.LOW)
            zaznamenano = 0
radiace = impulz/(x*60)
print("Celkem", impulz, "impulzu za", x, "min.")
print("Radiace:", round(radiace, 5), "pulsu/s")

GPIO.cleanup()

```

Posledním zdrojovým kódem, jenž zde uvádím, je ještě více upravený kód výše. Po spuštění se tento program zeptá na název souboru, do kterého se bude postupně zaznamenávat průběh měření. Po dokončení měření uloží program výsledek do dalšího souboru, do kterého se ukládají i další výsledky z dalších měření (pokud se nachází ve stejném adresáři). Pro přehlednost ještě zopakují, co ukazují proměnné: *timeout* (celková doba měření v sekundách), *predcas* (procesorový počet sekund na začátku měření), *impulz* (suma zaznamenaných impulsů), *zaznamenano* (logická proměnná zabraňující v cyklovaném zaznamenání jednoho impulsu), *aktradm* (okamžitý poměr impulsů a sekund), *aktradm* (okamžitý poměr impulsů a minut), *aktdose* (aktuální dávka v $\mu\text{Sv/h}$), *cas* (počet sekund od začátku měření), *pulsy* (nejpřesnější celkový počet pulsů za sekundu), *pulsym* (celkový počet impulsů převeden na pulsy za minutu), *radiace* (pulsy za minutu vynásobené kalibrační konstantou vyjadřující pohlcenou dávku v $\mu\text{Sv/h}$).

```

#!/usr/bin/python3
# -*- encoding: utf-8 -*-

import RPi.GPIO as GPIO
import time

# pin 7 --> GPIO 4
# pin 17 --> GPIO 11

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.OUT)
GPIO.setup(17, GPIO.IN)

```

```

x = float(input('Zadejte dobu mereni v minutach: '))
nazev = input('Zadejte název souboru, do kterého se uloží průběh měření
(bez přípony): ')
prubeh = open("%s.csv" % nazev, 'a')
prubeh.write("Impulz; Pulz/Sekunda; uSv/h; Pulz/Minuta; Sekunda měření; %
do konce\n")

timeout = time.time() + x*60
predcas = time.time()
impulz = int(0)
zaznamenano = 0

while (time.time() < timeout):
    if(GPIO.input(17) == False):
        GPIO.output(4, GPIO.HIGH)
        if(zaznamenano == 0):
            impulz = impulz + 1
            aktradm = round(impulz/(time.time()-predcas), 2)
            aktradm = round((impulz/(time.time()-predcas)*60), 2)
            aktdose = round((impulz/(time.time()-predcas)*60)*0.0063, 4)
            cas = round(time.time()-predcas, 1)
            print("Imp: %s; %s pulsu/s; %s uSv/h; %s pulsu/min; Čas:
%s/%ss" % (impulz, aktradm, aktdose, aktradm, cas, round(x*60, 0)))
            zlomcasu = round((time.time()-predcas)/(x*60), 3)
            prubeh.write("%s; %s; %s; %s; %s; %s\r\n" % (impulz, aktradm,
aktdose, aktradm, cas, zlomcasu))
            zaznamenano = 1
        else:
            GPIO.output(4, GPIO.LOW)
            zaznamenano = 0

pulsy = impulz/(x*60)
pulsym = impulz/x
radiace = (impulz/x)*0.0063
print("Celkem", impulz, "impulzu za", x, "min.")
print("Impulsy:", round(pulsy, 5), "pulsu/s,", round(pulsym, 5), "pulsu/m")
print("Dávka:", round(radiace, 7), "uSv/h")

print("Průběh měření zapsán do souboru %s.csv" % nazev)

vysledek = open("vysledky.txt", 'a')
vysledek.write(" %s: %s pulsy/min %s pulsy/s (%s uSv/h) s celkem %s pulsy
za %s min.\r\n" % (nazev, round(pulsym, 5), round(pulsy, 5), round(radiace,
7), impulz, x))
print("Výsledek zapsán do souboru vysledky.txt")
print("Hodnoty: Název; Pulsu/s; Pulsy/min; Dávka; Impulzy celkem; Čas
měření v minutách")

prubeh.close()
vysledek.close()
GPIO.cleanup()

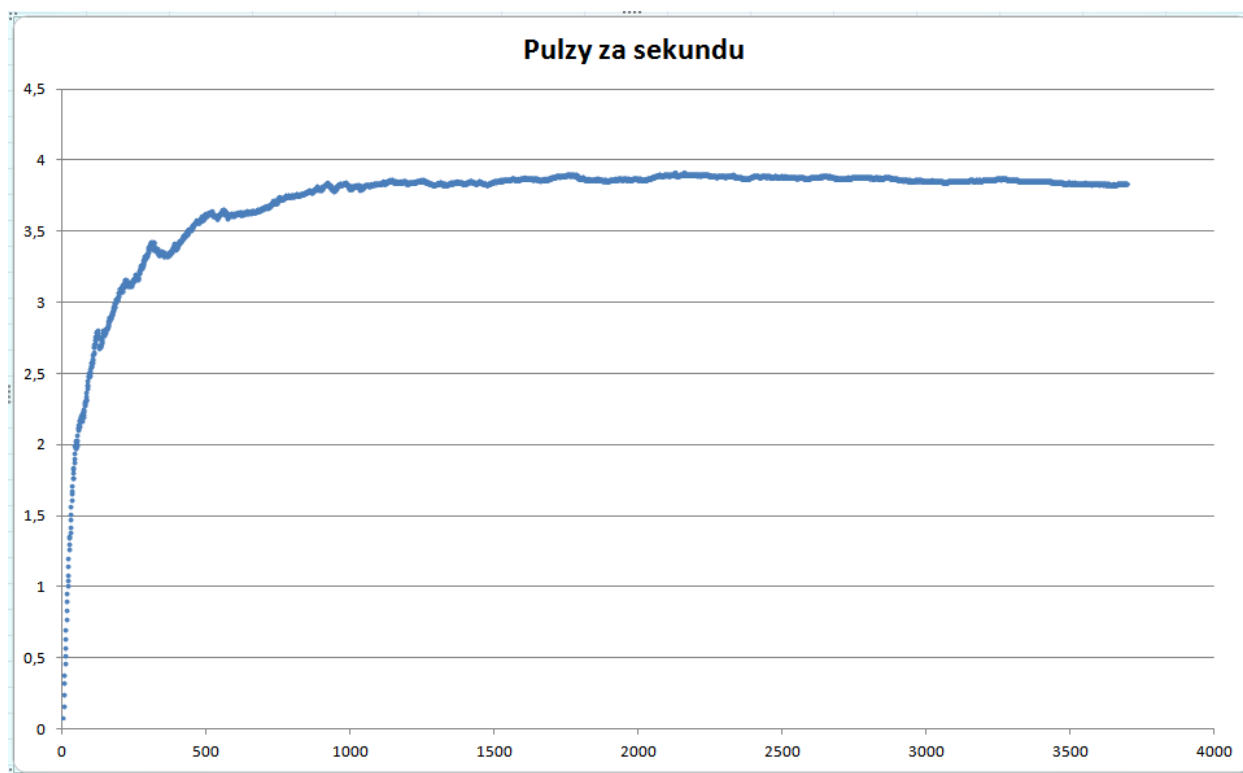
```

3.7 Výpis dat

Existuje mnoho možností, jak zobrazit získaná data. Od vypsaní v textovém souboru, přes sdílení na internetových stránkách, až po přidání na databázi, pokud by se měřič velmi využíval. Zatím jsem se rozhodl využít pouze tři metod.

Úplně nejjednodušší metodou je výpis výsledků do textového souboru. Tento výpis poskytuje výhody snadného programování, možnosti snadného čtení ze souboru a možnosti vlastního porovnání. Nicméně ve velkém množství výsledků se nemusí každý vyznat, navíc nejsou zcela vhodné k veřejnému publikování.

Druhou více názornou metodou je využití tabulkového programu s funkcemi grafů (např. MS Excel). Tato metoda umožňuje zobrazit průběh měření i porovnávat výsledky. Z hlediska programování je jednoduchá stejně jak metoda výpisu do textového souboru. Při výpisu velkého množství dat do textového souboru se jednotlivá data jednoho prvku oddělí středníkem a následující data dalšího prvku dalším řádkem. Pro MS Excel musí mít výsledný textový soubor příponu `.csv` k okamžitému spuštění.



Třetím nástrojem je technologie Google Maps Java Script API umožňující navrhnout si vlastní mapu obsahující vlastní body, obrazce, značky aj. pomocí jazyka Java Script. Výhodou je její snadná a moderní možnost internetové publikace. K získání kódu API je potřeba mít Google účet. Mapa lze jednoduše zobrazit na stránce HTML s odkazem na Java Script:


```

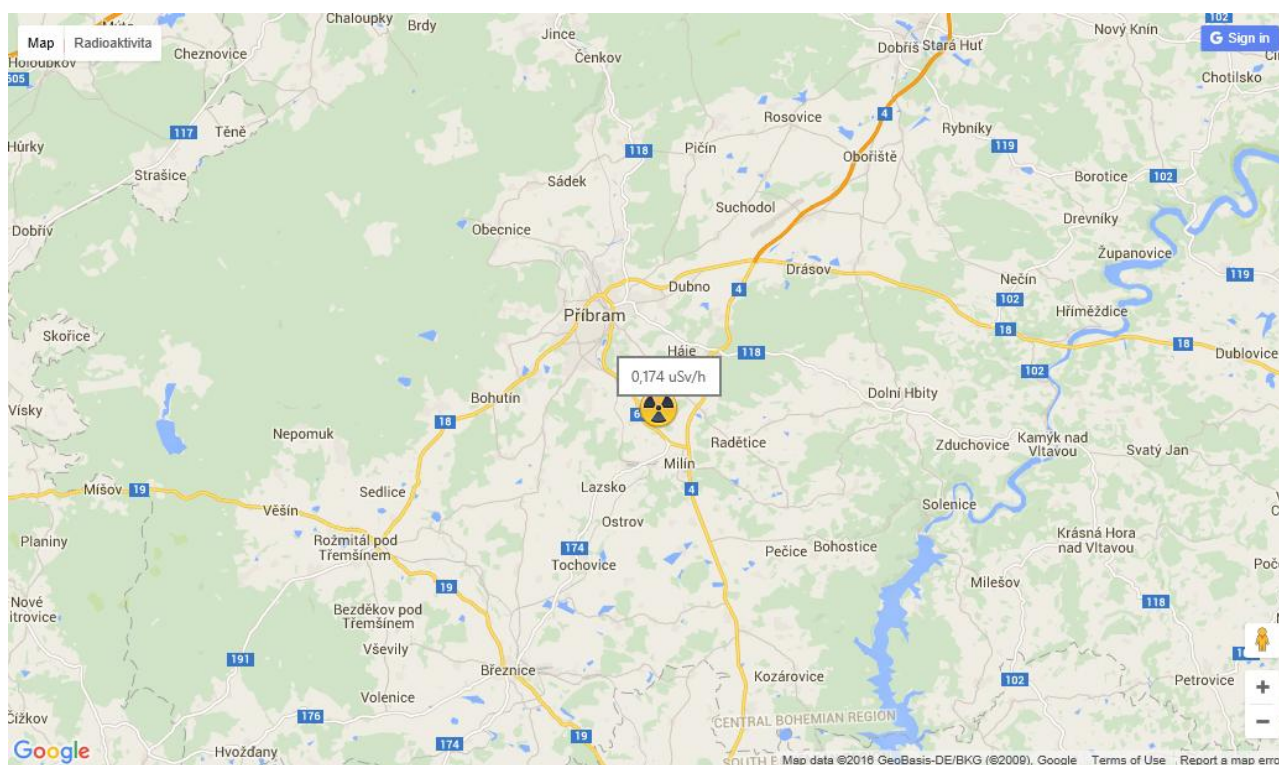
function initMap() {
var myLatLng = {lat: -25.363, lng: 131.044};

var map = new google.maps.Map(document.getElementById('map'), {
  zoom: 4,
  center: myLatLng
});

var marker = new google.maps.Marker({
  position: myLatLng,
  map: map,
  title: 'Hello World!'
});
}
<script src="mapa.js"> </script>
<script async defer
  src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">
</script>

```

Pokud by RasPi měl možnost získávat GPS souřadnice, mohl by po každém měření umístit na mapu tyto body s hodnotami dávky. Po podrobnějším vývoji kódu mohou mapy vypadat následovně:



Prozatím, na adrese <http://81.0.245.114/rad> jsou online k dispozici mapy s mnou naměřenými hodnotami.

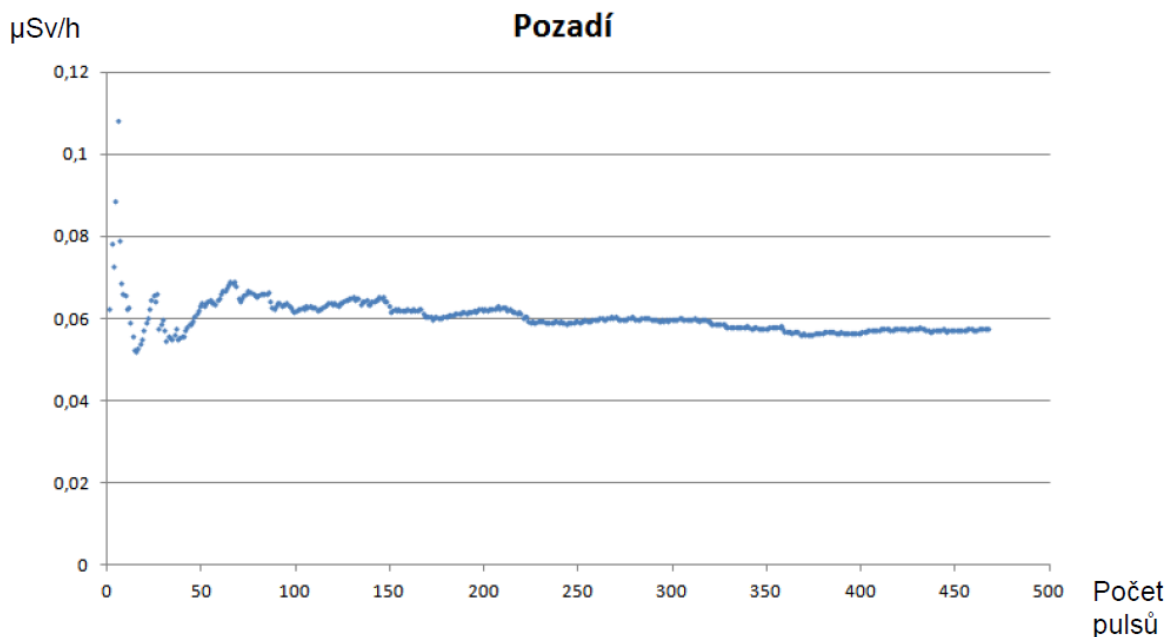


3.8 Výsledky vlastní měření radioaktivity vybraných předmětů

Nejdříve je nutno zmínit, že v tvarovacím obvodu jsem kvůli možnému nízkému maximálnímu počtu impulsů za sekundu vyměnil rezistor R1 s odporem 187 k Ω za rezistor s 30 k Ω . Doba signálu 555 vychází na 0,73 ms, a z toho vychází počet možných signálů k zaznamenání přibližně na 1380 impulsů za sekundu.

Předměty byly měřeny dvěma přístroji: vlastním měřičem radioaktivity trubicí STS-5 a měřičem GAMMA CHECK - tímto měřičem bylo možno měřit maximálně 180 sekund.

Měřeno bylo uranové sklo, UV záření z rtuťové výbojky, wolframová elektroda a demonstrační zářič Am 241. Při měření byl zaznamenáván poměr impulsů vzhledem k uplynulému času od začátku měření a celkový výsledek dávky (viz obrázky 9, 10, 11 v příloze). Uranové sklo a demonstrační zářič byly měřeny ze vzdálenosti 10 cm a z co největší blízkosti (sklo zevnitř)



Z grafu můžeme vidět, že měření je zpočátku velmi nepřesné a později je více přesnější. Nejvíce je tento jev vidět na pozadí s nejmenším počtem impulsů v poměru s časem oddalujícího se od nuly.

STS-5

	Pulsy/min	δ Pulsy/min	$\mu\text{Sv/h}$	Celkem pulsů	Bez pozadí	Pozadí	Doba m.
Wolframová elektroda	16,311	$\pm 0,43$	0,1028	1468	-23	1491	90 min.
Rtuťová výbojka	22,6	$\pm 0,87$	0,1424	678	211	467	30 min.
Americium 241 10 cm	78,67	$\pm 1,62$	0,4956	2360	1743	617	30 min.
(z těsné blízkosti)	1100,2	$\pm 14,83$	6,9313	5501	5398	103	5 min.
Uranové sklo 10 cm	120,5	± 2	0,7592	3615	3114	501	30 min.
(z těsné blízkosti)	1650,4	$\pm 18,71$	10,3975	8252	8170	84	5 min.

GAMMA CHECK

	Pulsy/min	δ Pulsy/min	$\mu\text{Sv/h}$	Celkem pulsů
Americium 241 10 cm	35,3	$\pm 3,43$	1,185	106
(z těsné blízkosti)	345,7	$\pm 10,73$	12,095	1037
Uranové sklo 10 cm	10,3	$\pm 1,86$	0,52	31
(z těsné blízkosti)	21,7	$\pm 2,69$	0,72	65

3.9 Výsledky vlastního měření radioaktivity v určitých lokalitách ČR

Moje pozornost byla zaměřena na tři různá místa a to podle počtu obyvatel v dané lokaci, nadmořské výšky a těžby nerostného materiálu:

- Praha
- Jizerské hory
- Slivice (u Příbramy)

Praha byla měřena na dvou místech: na Barrandově a v Bohnicích. Na Barrandově je průměrná ekvivalentní dávka **0,098 $\mu\text{Sv/h}$** , v Bohnicích **0,12 $\mu\text{Sv/h}$** . Počet obyvatel je v této oblasti vysoký, průměrná nadmořská výška je 288 m n. m.

V **Jizerských horách** v obci Horní Maxov bylo naměřeno **0,177 $\mu\text{Sv/h}$** . Měření proběhlo v nadmořské výšce 600 m n. m. a dříve tu byly cínové doly.

Ve vesničce **Slivice** bylo naměřeno **0,296 $\mu\text{Sv/h}$** . Nadmořská výška tu je 573 m n. m. a nedávno se tu těžil uran. Přibližně 7 km od místa měření se nachází uranové doly, dříve otevřené pro návštěvu turisty.

ZÁVĚR A DISKUZE

Výsledky byly očekávané i neočekávané. Při měření předmětů jsem očekával nejvíce radioaktivní laboratorní zářič s americiem, což potvrzoval měřič GAMMA CHECK. Nicméně toto můj detektor nepotvrdil. Očekával jsem také větší radioaktivitu u wolframových elektrod. Po odečtení dávky pozadí vyšly dokonce i záporné hodnoty, takže radioaktivita wolframové elektrody se 3 % oxidu thoria není potvrzena. Při vlastním měření uranového skla jsem naměřil značně větší dávku než pomocí měřiče GAMMA CHECK. Hlavní rozdíl mezi zářením uranového skla a americia 241 spočívá v energii záření. Energie γ rozpadu americia 241 je 60 keV. Pokud budeme hledat γ rozpadající se izotopy obsahující uranové sklo, můžeme zde nalézt uran 235 s energií γ rozpadu 186 keV a thorium 234 s energií γ rozpadu 63 keV až 93 keV. Trubice STS-5 je tedy citlivější na tvrdší γ záření, než trubice v měřiči GAMMA CHECK.

Při měření prostředí byl potvrzen zvýšený ekvivalent v obci Slivice (poblíž uranových dolů). Nicméně tato dávka nepřesahuje hranice bezpečnosti, a tak dlouhodobý pobyt v oblasti je ještě bezpečný.

Tato práce shrnuje základní poznatky související s radioaktivitou, dozimetrií, elektronikou, počítačem Raspberry Pi a alternativními softwary. Kombinace těchto poznatků jsem se pokusil využít k sestavení měřiče radioaktivity, jenž poskytuje mnoho možností k propojení s dalšími technologiemi.

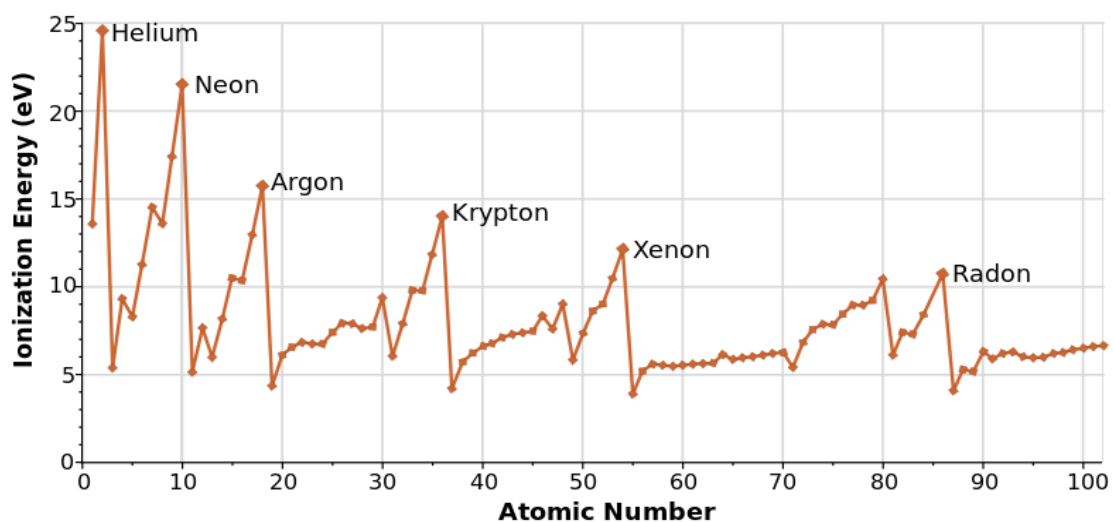
Jednoduchost, názornost a nízká cena jsou obrovskými výhodami tohoto projektu. Vždyť celková cena součástek pro detektor včetně ceny Raspberry Pi byla přibližně 1400 Kč. Této výhody by mohly využít dnešní střední školy díky nízké ceně a obrovskému počtu možností, jak Raspberry Pi využít (např. ovládání robotů, pohybový senzor, ovládání výtahu, programování, učení základu Linuxu, ...)

LITERATURA

- [1] ULLMAN V. *Jaderná a radiační fyzika* [online]. Dostupné z URL:
<http://astronuklfyzika.cz/JadRadFyzika2.htm>
- [2] ŠTOLL I. *Fyzika pro gymnázia – Fyzika mikrosvěta*. Praha: Prometheus, 1993, s. 104 - 107.
ISBN 8085849488
- [3] NORRIS D. *Raspberry Pi*. Brno: Computer Press, s. 232 - 234. ISBN 8025143465
- [4] ULLMAN V. *Jaderná a radiační fyzika* [online]. Dostupné z URL:
<http://astronuklfyzika.cz/JadRadFyzika6.htm>
- [5] VALÁŠEK J. *Detekce ionizujícího záření* [online]. Dostupné z URL:
<http://www.radioaktivita.cz/jednotky.html>
- [6] VERITASIAM. *The Most Radioactive Places on Earth*. In: Youtube [online]. Zveřejněno 16. 04. 2012 [vid. 2014-12-17]. Dostupné z URL:
<https://www.youtube.com/watch?v=TRL7o2kPqw0>
- [7] TRANSLATORSCAFE. *Unit Converter* [online]. Dostupné z URL:
<http://www.translatorscafe.com/cafe/EN/units-converter/radiation/c/>
- [8] WELSMORE E. *The Plutonium Files, America's Secret Medical Experiments in the Cold War*. New York: Delta Publishing. ISBN 0307767377
- [9] WIKISKRIPTA *Ionizace* [online]. Dostupné z URL:
<http://www.wikiskripta.eu/index.php/Ionizace> <http://www.wikiskripta.eu/index.php/Ionizace>
- [10] RECHL J. *Encyklopedie fyziky* [online]. Dostupné z URL:
<http://fyzika.jreichl.com/main.article/view/849-proporcionalni-komora>
- [11] DANYK *Kapesní detektor radioaktivity* [online]. Dostupné z URL: <http://www.danyk.cz>
- [12] EMBEDDED LINUX WIKI *RPi Hardware* [online]. Dostupné z URL:
http://elinux.org/RPi_Hardware
- [13] RASPBERRY PI FOUNDATION *Raspberry Pi 3 Model B* [online]. Dostupné z URL:
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [14] WIKIPEDIA *Raspberry Pi* [online]. Dostupné z URL:
https://cs.wikipedia.org/wiki/Raspberry_Pi
- [15] EMBEDDED LINUX WIKI *RPi Low-level peripherals* [online]. Dostupné z URL:
http://elinux.org/RPi_Low-level_peripherals
- [16] WIKIPEDIA *EXE* [online]. Dostupné z URL: <https://cs.wikipedia.org/wiki/EXE>
- [17] WIKIPEDIA *Python* [online]. Dostupné z URL: <https://cs.wikipedia.org/wiki/Python>
- [18] PETERKA J. *Compiler vs. interpreter* [online]. Dostupné z URL:
<http://www.earchiv.cz/a95/a506c120.php3>

- [19] BÍLEK P. *Úvod do jazyka Python* [online]. Dostupné z URL:
<http://www.sallyx.org/sally/python/python1.php>
- [20] PYTHON SOFTWARE FOUNDATION *What's New In Python 3.0* [online]. Dostupné z URL: <https://docs.python.org/3/whatsnew/3.0.html>
- [21] Stackoverflow: *Purpose of #!/usr/bin/python3* [online]. Dostupné z URL:
<http://stackoverflow.com/questions/7670303/purpose-of-usr-bin-python3>
- [22] NORRIS D. *Raspberry Pi*. Brno: Computer Press, s. 41 - 49. ISBN 8025143465
- [23] NORRIS D. *Raspberry Pi*. Brno: Computer Press, s. 59 - 61. ISBN 8025143465
- [24] WIKIPEDIA *NE555* [online]. Dostupné z URL: <https://cs.wikipedia.org/wiki/NE555>

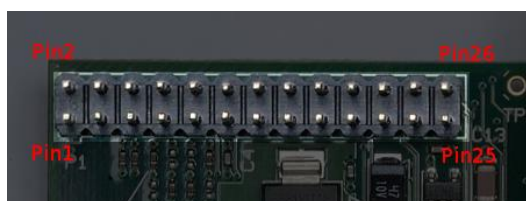
PŘÍLOHY



OBR. 1 IONIZAČNÍ ENERGIE PRVKŮ [9]



OBR. 2 RASPBERRY PI 1, MODEL B



OBR. 4 GPIO KONEKTOR

	Pin 1	Pin 2		Pin 25	Pin 26
+3V3	Orange square	Red circle	+5V	Green circle	Black circle
GPIO2 / SDA1	Cyan circle	Red circle	+5V	Green circle	Black circle
GPIO3 / SCL1	Cyan circle	Black circle	GND	Green circle	Black circle
GPIO4	Green circle	Yellow circle	TXD0 / GPIO 14	Green circle	Black circle
GND	Black circle	Yellow circle	RXD0 / GPIO 15	Green circle	Black circle
GPIO17	Green circle	Green circle	GPIO 18	Green circle	Black circle
GPIO27	Green circle	Black circle	GND	Green circle	Black circle
GPIO22	Green circle	Green circle	GPIO 23	Green circle	Black circle
+3V3	Orange square	Green circle	GPIO 24	Green circle	Black circle
GPIO10 / MOSI	Purple circle	Black circle	GND	Green circle	Black circle
GPIO9 / MISO	Purple circle	Green circle	GPIO 25	Green circle	Black circle
GPIO11 / SCLK	Purple circle	Purple circle	CE0# / GPIO8	Green circle	Black circle
GND	Black circle	Purple circle	CE1# / GPIO7	Green circle	Black circle

OBR. 3 VÝVODY KONEKTORŮ GPIO



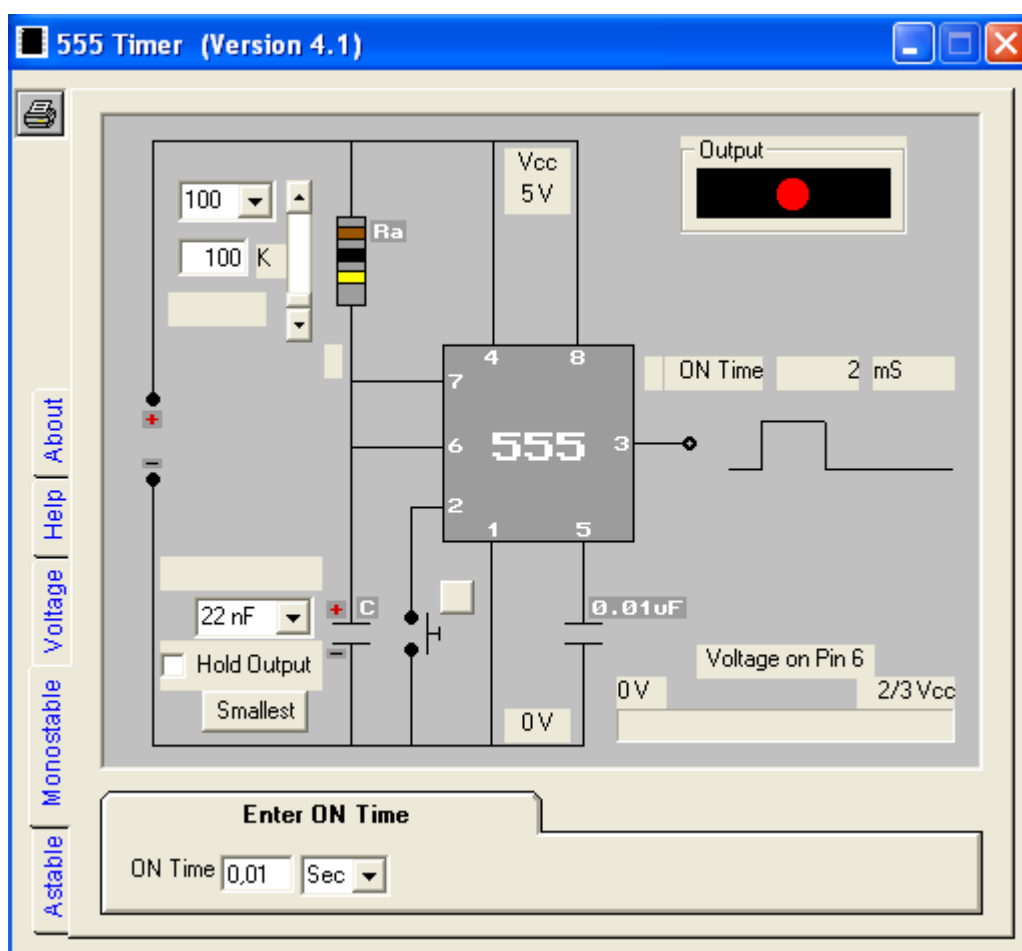
OBR. 5 TRUBICE STS-5



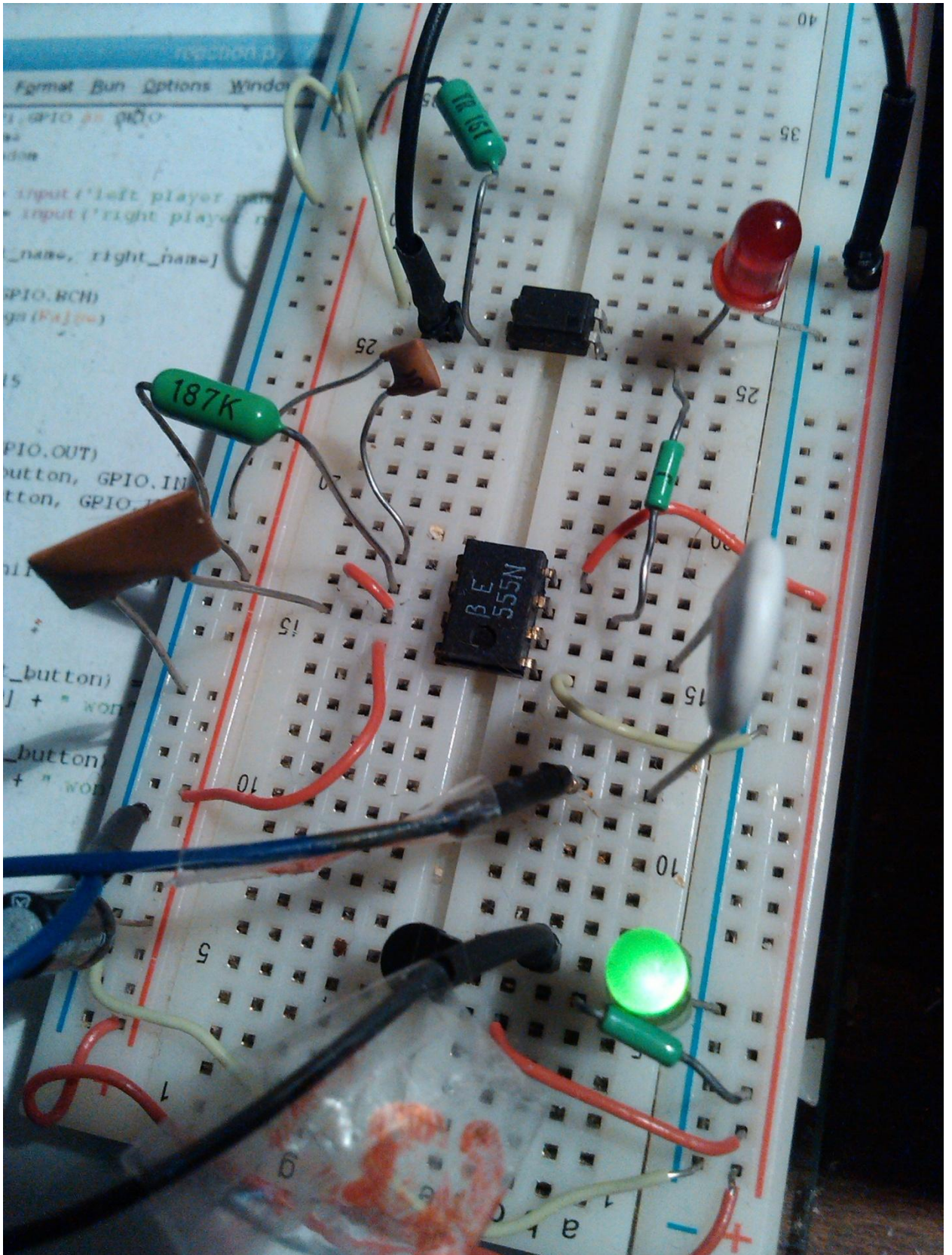
OBR. 6 ČASOVAČ 555



OBR. 7 OPTRON



OBR. 8 PROGRAM PRO SCHÉMA AVÝPOČET ČASOVAČE 555



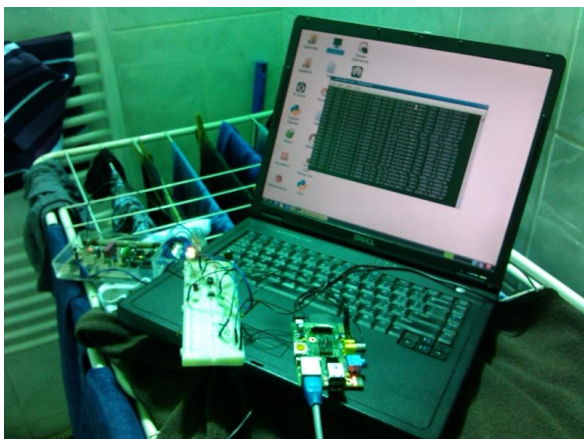
OBR. 9 ZAPOJENÍ NA KONTAKTNÍM POLI



OBR. 10 VÁZA Z URANOVÉHO SKLA



OBR. 11 RTUŽOVÁ VÝBOJKA



OBR. 12 MĚŘENÍ UV ZÁŘENÍ RTUŽOVÉ VÝBOJKY



OBR. 13 MĚŘENÍ AKTIVITY VÁZY Z URANOVÉHO SKLA ZE VZDÁLENOSTI 5 CM



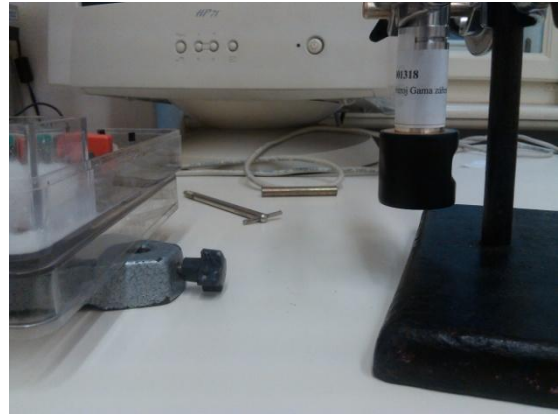
OBR. 14 MĚŘENÍ URANOVÉHO SKLA ZE VZDÁLENOSTI 10 CM



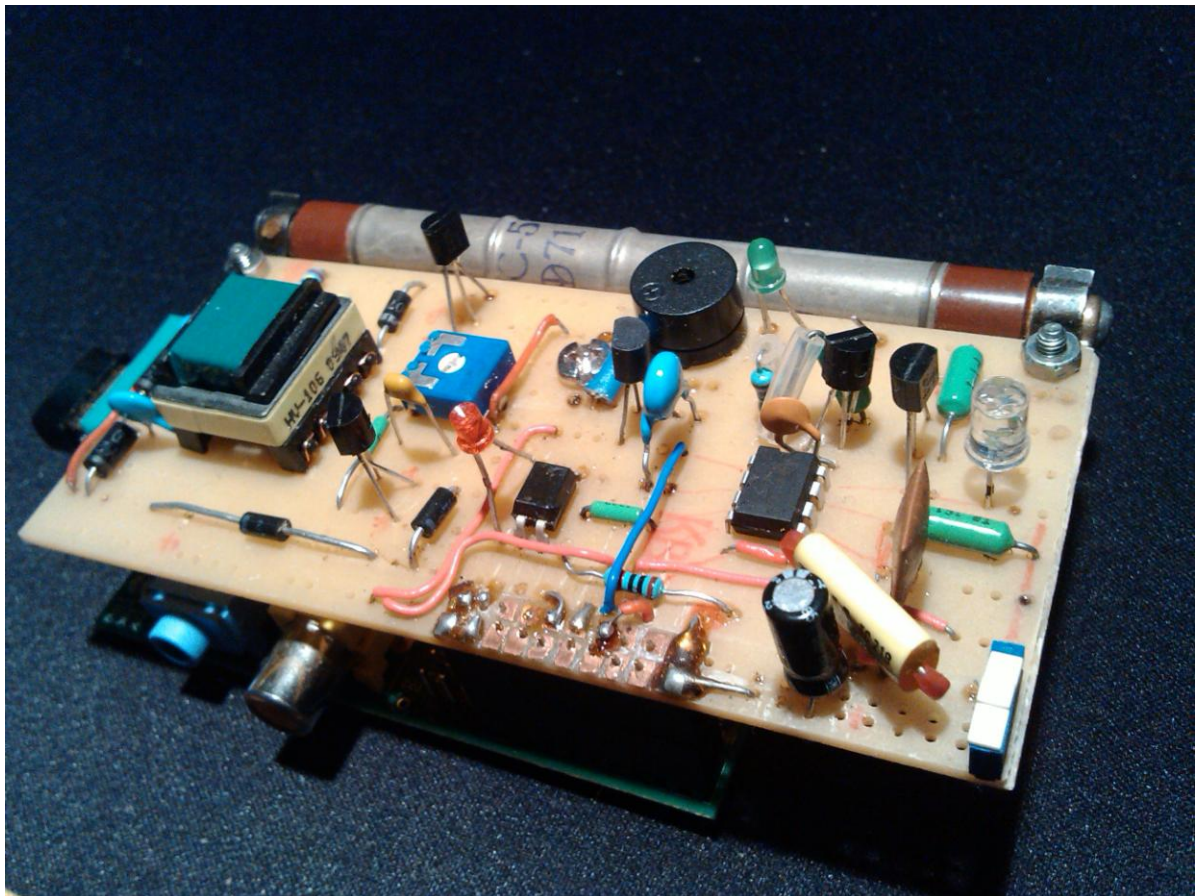
OBR. 15 MĚŘENÍ URANOVÉHO SKLA Z TĚSNÉ BLÍZKOSTI



OBR. 16 MĚŘENÍ URANOVÉHO SKLA Z 10 CM
POMOCÍ WOLFCRAFT GAMMA CHECK



OBR. 17 MĚŘENÍ AMERICIA Z 10 CM
POMOCÍ STS-5



OBR. 17 KOMPLETNÍ MĚŘIČ RADIOAKTIVITY

Datasheet trubice STS-5



А. Основные типовые параметры

1. Напряжение начала счета 280-330 в
2. Рекомендуемое рабочее напряжение 360-440 в
3. Наименьшая протяженность плато 80 в
4. Наибольший наклон плато 0,125% на 1 в
5. Наибольший натуральный фон 27 имп. в мин.

Б. Условия эксплуатации

1. Сопротивление нагрузки от 5 до 10 Мом
2. Наибольшая паразитная емкость, допустимая во входной цепи счетно-измерительного устройства 10 пф
3. Переходная емкость входа счетного прибора 7-15 пф
4. Допустимые колебания температуры окружающей среды от -40 до +50°С
5. Для обеспечения продолжительной и устойчивой работы счетчика следует:

а) не допускать попадания на счетчик всякого рода загрязнений, могущих ухудшить его изоляционные свойства;

б) оберегать счетчик от ударов и механических повреждений;

в) при включении счетчика соблюдать соответствующую полярность—положительный полюс источника питания присоединять к анодному выводу счетчика, обозначенному знаком «+», а отрицательный полюс источника питания присоединять к катодному выводу счетчика, не имеющему обозначения.

Штамп ОТК

110570