Zespołowy projekt studencki 2018/19
pod kierunkiem dr. hab. Krzysztofa Miernika

Osobisty detektor promieniowania z wykorzystaniem Arduino

Anna Daniluk, Anna Długołęcka, Anna Kwiatkowska,
Tomasz Rogiński, Monika Rykała

Streszczenie

Celem projektu było stworzenie osobistego detektora promieniowania jonizującego opartego na liczniku Geigera-Müllera (w skrócie liczniku G-M), z wykorzystaniem prostych kontrolerów i Arduino. Własnoręczne skonstruowanie urządzenia pozwoliło rozwinąć umiejętności programowania i konstrukcji prostych układów elektronicznych oraz pomogło lepiej zrozumieć zasadę działania licznika Geigera-Müllera. Detektor może zostać wykorzystany do monitorowania poziomu promieniowania jonizującego na terenie Wydziału Fizyki Uniwersytetu Warszawskiego albo stanowić element pokazów popularyzujących fizykę wśród uczniów.

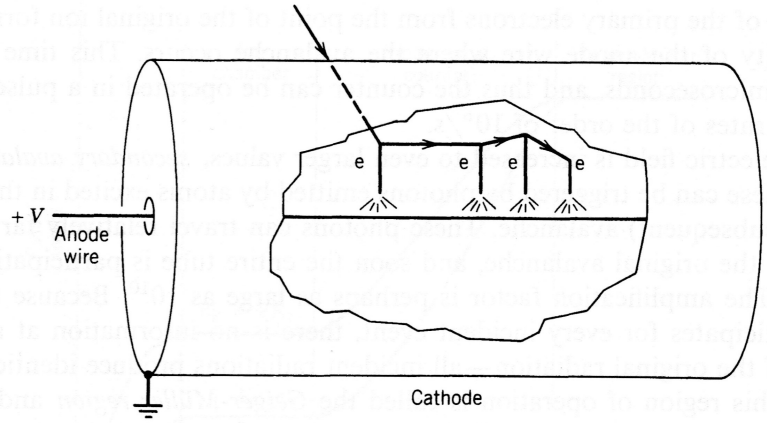
1 Wstęp

Monitorowanie poziomu promieniowania jonizującego jest kluczowym elementem ochrony radiologicznej, niezbędnym w każdym ośrodku wykorzystującym źródła promieniotwórcze. Ocena narażenia na promieniowanie jonizujące może być przeprowadzona z użyciem różnego rodzaju detektorów, zarówno pasywnych jak i aktywnych. Wśród tych ostatnich najpopularniejszym jest licznik Geigera-Müllera.

1.1 Budowa i zasada działania licznika Geigera-Müllera

Licznik Geigera-Müllera składa się z cylindrycznej rurki wypełnionej rozrzedzonym gazem. Wzdłuż jej osi, w centralnej części znajduje się cienki drut, zwykle wolframowy, stanowiący anodę i otoczony metalowym cylindrem będącym katodą. Do obu elektrod podłączone jest napięcie ok. 400 V. Działanie detektora G-M opiera się na rejestracji impulsów napięciowych powstających, gdy promieniowanie jonizujące oddziałuje z atomami gazu w tubie. Pojedynczy akt jonizacji powoduje powstanie pary kation-elektron (jonizacja pierwotna). Przy odpowiednim doborze napięcia zasilającego naładowane cząstki pochodzące z jonizacji pierwotnej mają energie wystarczające do tego, aby w zderzeniach z pozostałymi atomami gazu roboczego doprowadzić do wyładowania lawinowego obejmującego całą objętość gazu (patrz Rysunek 1). Elektrony, posiadające ujemny ładunek, poruszają się w kierunku anody, natomiast dużo wolniejsze jony przemieszczają się w stronę katody. Dzięki lawinowemu charakterowi wyładowania, wygenerowane impulsy napięciowe mają równe amplitudy. Tracona

jest informacja o energii padającego promieniowania jonizującego, jednak osiąga się większą czułość detektora. Po zastosowaniu odpowiedniego układu zliczającego możliwy jest pomiar liczby zarejestrowanych cząstek pierwotnych, które wywołały jonizację.



Rysunek 1: Schemat licznika G-M [1].

1.2 Platforma programistyczna Arduino

Licznik skonstruowany w ramach projektu jest układem elektronicznym opartym na platformie programistycznej Arduino (Open Hardware) [2].

Arduino to popularne moduły zawierające mikrokontrolery. Posiadają one bogate środowisko programistyczne (IDE), wykorzystują ogólnodostępne biblioteki i płytki rozszerzające, mają przyjazną składnię zbliżoną do języka C/C++. Główną część modelu Arduino Nano, wykorzystanego ze względu na mały rozmiar, stanowi mikrokontroler AVR. W skład podstawowego wyposażenia płytki wchodzą też m.in. 22 programowane wejścia/wyjścia cyfrowe, 8 wejść analogowych, wyjścia PWM, przycisk resetu, diody LED. Zasilanie oraz komunikacja z komputerem, w tym programowanie płytki, może odbywać się poprzez łącze miniUSB.

2 Schemat licznika Geigera-Müllera

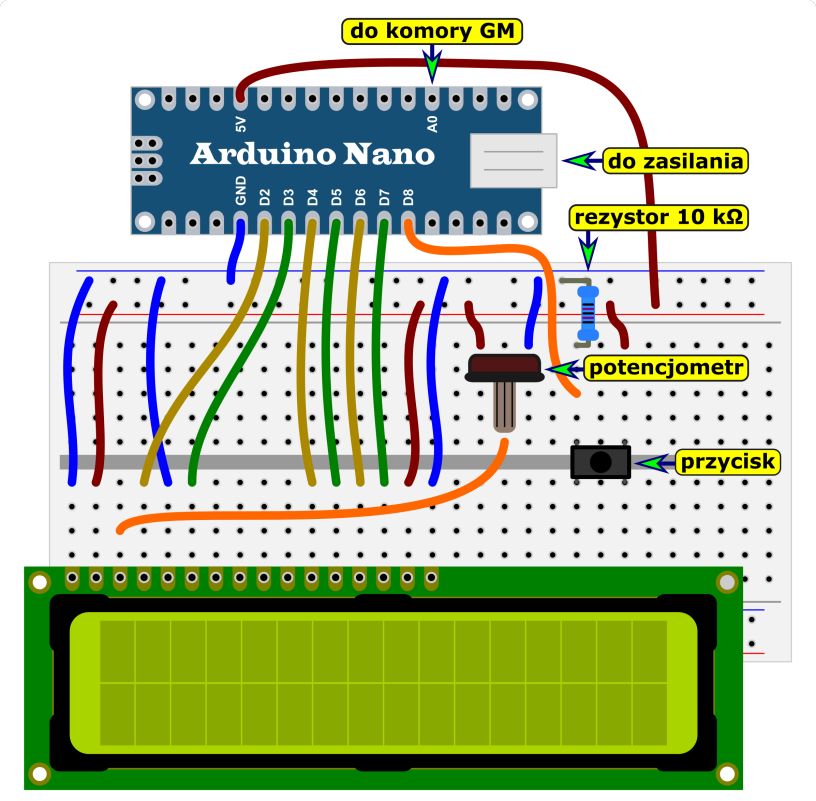
Do skonstruowania układu wykorzystano następujące elementy:

- tuba Geigera-Müllera STS-5,
- płyta Arduino Nano,
- przetwornica regulowana 5-12 V, 0,3-1,2 kV,
- alfanumeryczny wyświetlacz LCD 2×16,
- potencjometr B10K,
- rezystor metalizowany 10 MΩ, 0,5 W (2 sztuki),

- rezystor metalizowany 100 kΩ,
- kondensator ceramiczny 10 nF, 1000 V,
- dioda 1N4937,
- przycisk microswitch,
- kabel sieciowy miniUSB,
- zacisk uchwytu bezpiecznika, 6,35 mm - do zamocowania detektora G-M (2 sztuki),
- przewody połączeniowe męsko-męskie,
- płytka prototypowa (2 sztuki).

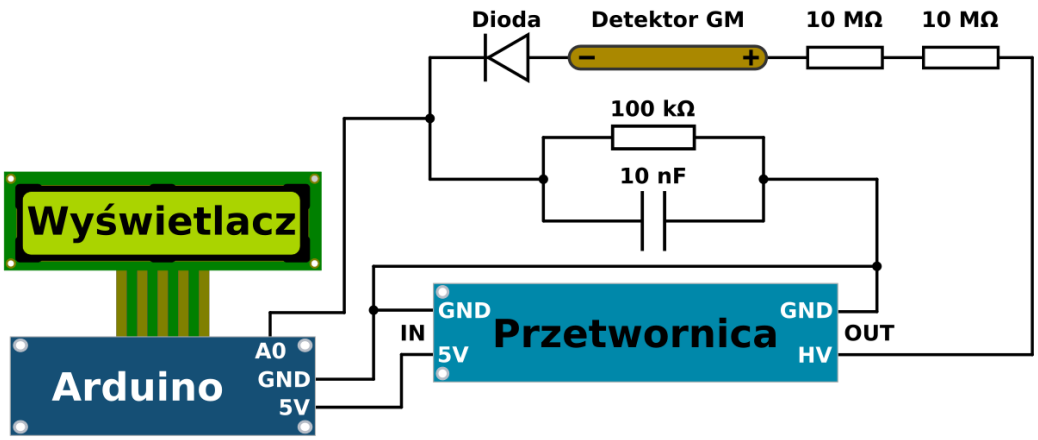
Płytę Arduino zaprogramowano korzystając z laptopa, natomiast do oglądania i oceny przebiegu sygnału użyto oscyloskopu.

Przed przystąpieniem do zmontowania właściwego układu, zawierającego detektor G-M, sprawdzono działanie wyświetlacza cyfrowego. Schemat podłączenia wyświetlacza do płytki Arduino przedstawiono na Rysunku 2. Płytkę zasilano z komputera poprzez łącze miniUSB. Rolą użytego potencjometru jest regulacja jasności wyświetlacza. Funkcja przycisku (z rezystorem) zostanie objaśniona w dalszej części pracy.



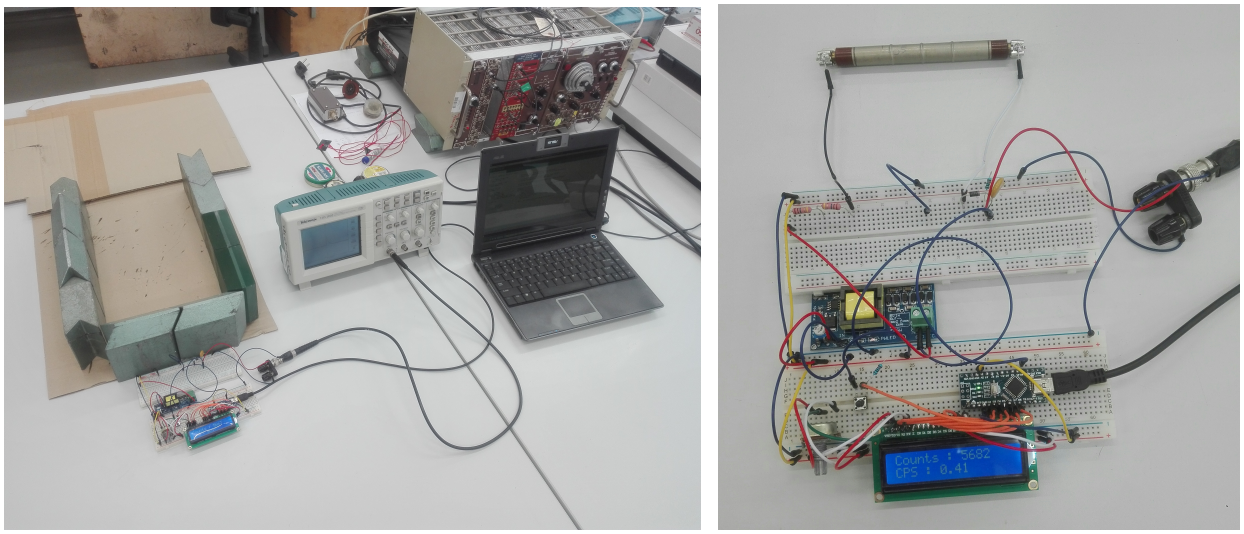
Rysunek 2: Schemat sposobu podłączenia wyświetlacza cyfrowego do płytki Arduino

Po weryfikacji poprawnego działania wyświetlacza zmontowano cały układ, zgodnie ze schematem pokazanym na Rysunku 3.



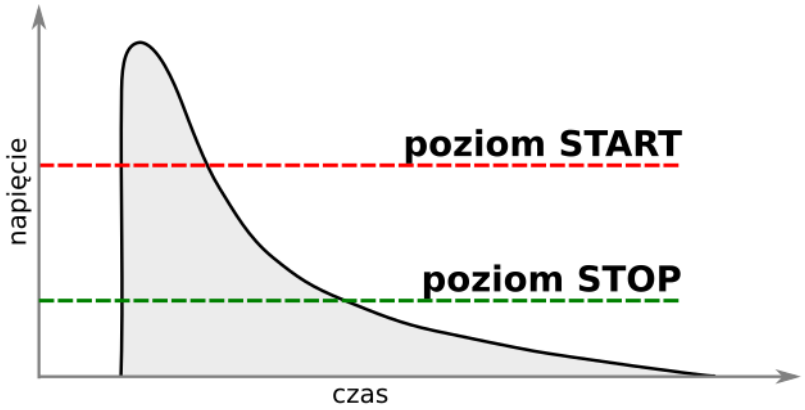
Rysunek 3: Schemat podłączenia detektora G-M do płytki Arduino. Symbole +, - oznaczają odpowiednio: katodę i anodę detektora.

Praktyczna realizacja uwzględniająca dodatkowo podłączenie do oscyloskopu i komputera została przedstawiona na Rysunku 4.



Rysunek 4: Realizacja całego układu.

Zasadniczą część układu stanowi detektor Geigera-Müllera. Jest on zasilany wysokim napięciem (zalecane napięcie to 360-440 V [3]), wytwarzanym przez przetwornicę, podłączoną do płyty Arduino. Między detektorem a przetwornicą umieszczono dwa oporniki o rezystancji 10 MΩ, zabezpieczające przed zbyt wysokimi wartościami natężenia prądu w układzie. Gdy w tubie G-M następuje wyładowanie lawinowe, przez diodę przepływa prąd, a na okładkach kondensatora gromadzi się ładunek elektryczny. Następnie ładunek ten odpływa poprzez rezystor 100 kΩ, czemu towarzyszy powstanie charakterystycznego impulsu napięciowego (patrz Rysunek 5) rejestrowanego przez Arduino (wejście analogowe A0).



Rysunek 5: Szkic sygnału powstającego przy wyładowaniu w komorze jonizacyjnej (objaśnienia w tekście)

Zastosowanie diody ma na celu wydłużenie czasu trwania tego impulsu, dzięki wykorzystaniu jej zdolności przewodzenia prądu tylko w jednym kierunku. Arduino przetwarza sygnał analogowy z zakresu 0-5 V na cyfrowy, przypisując mu wartość całkowitą z przedziału 0-1023, proporcjonalnie do amplitudy. W przypadku, gdy mierzony sygnał wykracza poza ustalony zakres napięcia, przypisywana jest mu skrajna wartość 0 lub 1023. Poprawność działania układu została zweryfikowana poprzez obserwację impulsu, generowanego przez detektor, na ekranie oscyloskopu.

Następnie płytka Arduino została zaprogramowana w sposób opisany w dalszej części pracy.

3 Objaśnienie kodu programu

Na Rysunkach 6 i 7 zamieszczono pełną treść kodu. Część pierwsza służy zadeklarowaniu wszystkich zmiennych i stałych użytych w kodzie, zaś część druga zawiera wstępne ustawienia wyświetlacza. Część 3A określa reakcję układu na wciśnięcie przycisku: wyzerowanie wskazania wyświetlacza i odświeżenie jego ekranu. Część 3B stanowi najważniejszą sekcję kodu, ponieważ wyznacza ona sposób, w jaki płytka Arduino identyfikuje sygnał napięciowy powstały wskutek wyładowania w komorze jonizacyjnej. W każdym przebiegu pętli sprawdzane jest, czy sygnał na wejściu A0 przekracza wartość 80 (we wspomnianej wcześniej skali 0-1023) – poziom START (patrz Rysunek 5). W takim przypadku uznajemy, że odbierany jest impuls pochodzący od cząstki jonizującej. W momencie gdy wysokość sygnału spadnie poniżej wartości 30 (poziom STOP), układ zwiększa liczbę zliczeń o 1. W części 3C następuje wyznaczenie liczby zliczeń zarejestrowanych w kolejnych oknach czasowych, których szerokość (czas trwania) w sekundach jest określona wartością zmiennej `szerokosc_okna` oraz uwzględnienie wydajności detekcji.

```
#include <LiquidCrystal.h>                    // Dołączenie biblioteki

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);          // Informacja o podłączeniu nowego wyświetlacza

// Część 1:  Deklaracja zmiennych i stałych  //

const int pin_przycisku = 8;                  // Informacja, że przycisk jest podłączony
//                                               do wejścia D8
int stan_przycisku = 0;                       // Zmienna przechowująca stan przycisku
(HIGH oznacza przycisk wciśnięty)

const int pin_sygnalu = A0;                   // Informacja, że do wejścia A0 podłączony
//                                               jest sygnał analogowy
int stan_sygnalu = 0;                         // Zmienna przechowująca wartość mierzonego
//                                               napięcia

unsigned long czas;                           // Zmienna przechowująca czas [w milisekundach]
//                                               od uruchomienia detektora
int czekaj = 0;                               // Zmienna pomocnicza

long int zliczenia = 0;                       // Zmienna przechowująca liczbę zliczeń od chwili
//                                               ostatniego zresetowania detektora
long int numer_okna = 0;                      // Zmienna przechowująca numer bieżącego okna
//                                               czasowego
int szerokosc_okna = 5;                       // Zmienna wyznaczająca czas trwania jednego okna
//                                               czasowego [w sekundach]
int zliczenia_w_poprzednim_oknie = 0;         // Zmienna przechowująca wartość liczby zlicze,
//                                               ktore nastapily w poprzednim oknie czasowym
long int zliczenia_po_zamknieciu_okna = 0;    // Zmienna przechowująca wartość liczby zliczeń,
//                                               które nastąpiły od momentu ostatniego
//                                               zresetowania detektora do momentu końca
//                                               ostatniego okna czasowego
float CPS = 0;                                // Zmienna przechowująca wartość liczby zliczeń
//                                               na sekundę uśredniona dla ostatniego
//                                               okna czasowego

const float a0 = 5.0875;                      // a0, a1, a2 - współczynniki dopasowania
//                                               krzywej wydajności
const float a1 = 0.16564;
const float a2 = 0.01;

//------------ Koniec części 1 --------------//

//       Część  2: ustawienia wstępne         //

void setup() {
  lcd.begin(16, 2);                           // Deklaracja wymiarów wyświetlacza
  lcd.setCursor(0, 0);                        // Ustawienie kursora
  lcd.print("Counts : 0      ");              // Wyświetlenie tekstu
  lcd.setCursor(0, 1);
  lcd.print("                ");
}

//------------ Koniec części 2 --------------//
```

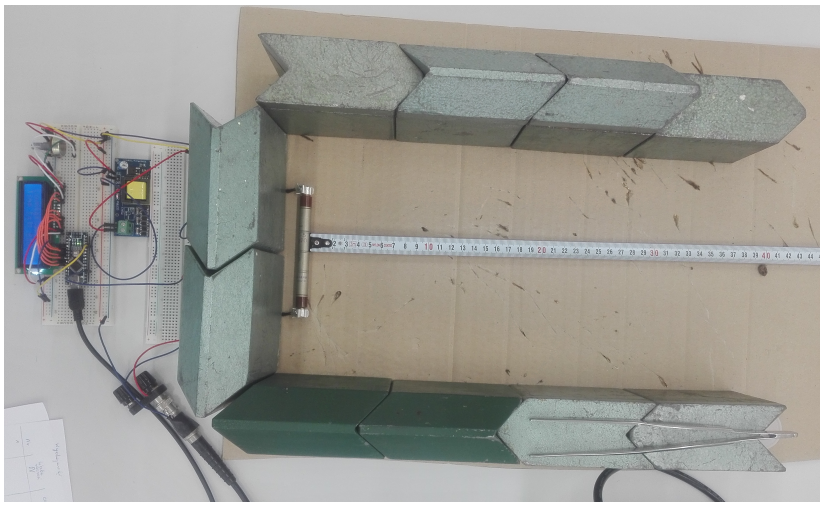
Rysunek 6: Kod programu - części: 1 i 2.

```
//     Część 3: właściwa część programu      //

void loop()
{

  // Część  3A: sprawdzenie, czy przycisk jest wciśnięty;
  // jeśli tak, następuje zresetowanie wskazania licznika
  stan_przycisku = digitalRead(pin_przycisku);
  if (stan_przycisku == HIGH)
  {
    zliczenia = 0;
    zliczenia_w_poprzednim_oknie = 0;
    zliczenia_po_zamknieciu_okna = 0;
    lcd.setCursor(0, 0);
    lcd.print("Counts : ");
    lcd.print(zliczenia);
    lcd.print("          ");
  }
  // Koniec części 3A

  // Część  3B: sprawdzenie, czy detektor zarejestrował zdarzenie
  // Odczytanie napięcia z wejścia analogowego
  stan_sygnalu = analogRead(pin_sygnalu);

  // Jeżeli sygnał analogowy przekracza wartość 80, zaczekaj,
  // aż spadnie poniżej wartości 30 i wtedy zwiększ liczbę
  // zliczeń o 1 oraz odśwież wskazanie wyświetlacza
  if (stan_sygnalu > 80)
  {
    czekaj = 1;
  }
  if (stan_sygnalu < 30 and czekaj == 1)
  {
    czekaj = 0;
    zliczenia += 1;
    lcd.setCursor(0, 0);
    lcd.print("Counts : ");
    lcd.print(zliczenia);
    lcd.print("          ");
  }
  // Koniec części 3B

  // Część  3C: obliczenie liczby zliczeń na sekundę
  // Sprawdź, czy w obecnej iteracji rozpoczęło się
  // kolejne okno czasowe; jeśli tak, wyznacz
  // liczbę zliczeń w poprzednim oknie czasowym
  czas = millis();
  if (czas > numer_okna * szerokosc_okna * 1000)
  {
    numer_okna += 1;
    zliczenia_w_poprzednim_oknie = zliczenia - zliczenia_po_zamknieciu_okna;
    zliczenia_po_zamknieciu_okna = zliczenia;
  }

  // Oblicz i wyświetl liczbę zliczeń na sekundę
  lcd.setCursor(0, 1);
  lcd.print("CPS : ");
  CPS = float(zliczenia_w_poprzednim_oknie) / szerokosc_okna;
  CPS *= (1 + pow(exp((CPS - a0) / a1), a2));
  lcd.print(CPS);
  lcd.print("       ");
  // Koniec części 3C

}

//------------ Koniec części  3 --------------//
```

Rysunek 7: Kod programu - część 3.

4 Kalibracja wydajnościowa

Przygotowano układ do przeprowadzenia pomiarów kalibracyjnych (patrz Rysunek 8).



Rysunek 8: Zdjęcie układu gotowego do kalibracji wydajnościowej

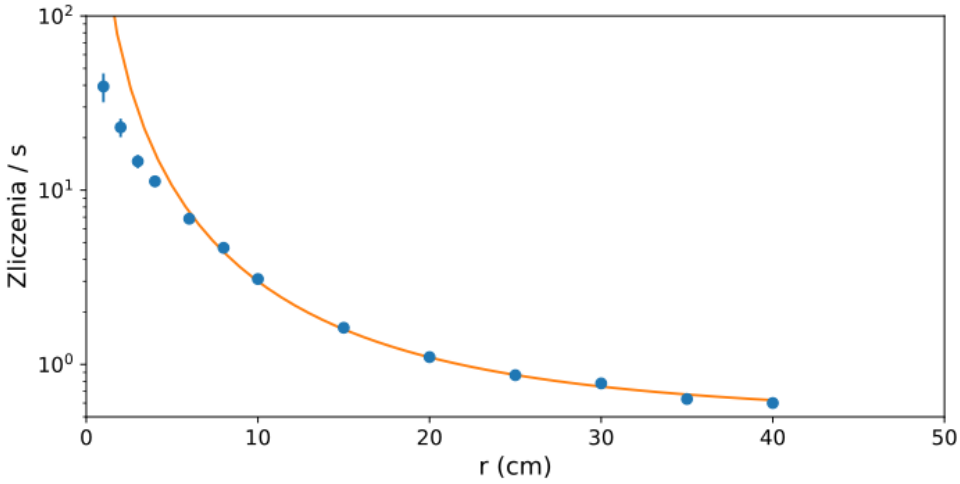
Zmierzono poziom promieniowania tła, uzyskując w czasie 97067 sekund (∼27 h) liczbę zliczeń równą 46173 ± 215, co odpowiada wartości (0,465 ± 0,003) cps. Za osłoną ołowianą umieszczono tubę G-M i źródło promieniowania. Dla 13 różnych odległości źródła od środka detektora zmierzono liczbę zliczeń w czasie odpowiadającym przynajmniej 400 zliczeniom. Wyniki pomiarów zebrano w Tabeli 1 poniżej. Przyjęto niepewność pomiaru czasu równą 1 s.

Tabela 1: Wyniki pomiarów dotyczących kalibracji wydajnościowej

Nr	Odległość r [cm]	Liczba zliczeń	Czas [s]	Liczba zliczeń na sekundę
1	40	402	670	0,60 ± 0,03
2	35	404	638	0,63 ± 0,04
3	30	413	531	0,78 ± 0,04
4	25	410	473	0,87 ± 0,05
5	20	403	366	1,10 ± 0,06
6	15	417	257	1,62 ± 0,08
7	10	445	144	3,09 ± 0,16
8	8	406	87	4,67 ± 0,26
9	6	418	61	6,85 ± 0,41
10	4	427	38	11,24 ± 0,81
11	3	410	28	14,6 ± 1,3
12	2	413	18	22,9 ± 2,8
13	1	433	11	39,4 ± 7,4
14	0	470	8	59 ± 15

Przyjęto, że niepewność pomiaru odległości jest równa rozmiarowi źródła, czyli 1 cm.

Wyniki pomiarów liczby zliczeń na sekundę w funkcji odległości r od źródła promieniowania przedstawiono na Rysunku 9. Poza punktami pomiarowymi, znajduje się na nim krzywa dopasowana metodą najmniejszych kwadratów.



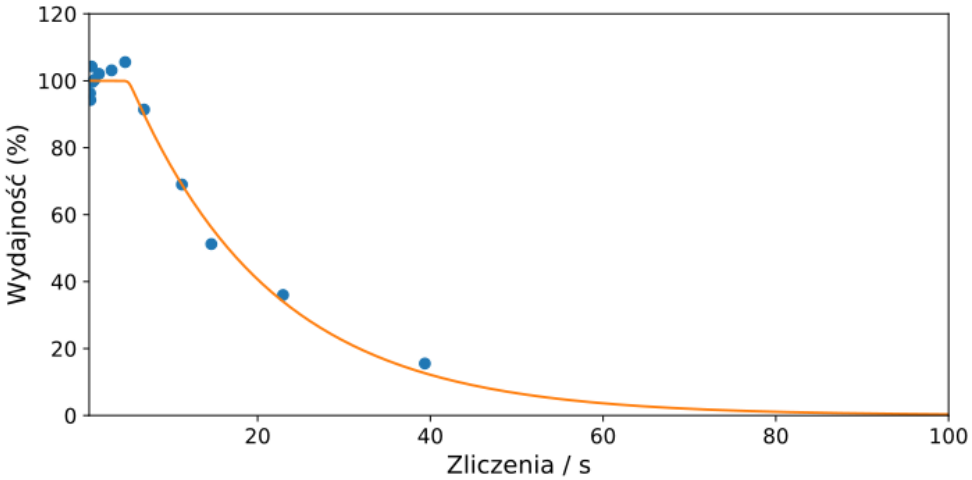
Rysunek 9: Zależność liczby zliczeń na sekundę od odległości od źródła promieniowania

Przy niewielkiej liczbie zliczeń na sekundę układ punktów na wykresie można opisać przy pomocy funkcji:

$$y(x) = a_0 + \frac{a_1}{x^2}. \tag{1}$$

Parametry dopasowania wynoszą: $a_0 = 0{,}465$ cps i $a_1 = (253,18 \pm 8,32)$ cps $\cdot$ cm^2 . Paramter a_0 uwzględnia promieniowanie tła. Punkty pomiarowe dla odległości $r < 6$ cm zostały pominięte w analizie, ponieważ dla tak małych odległości źródło-detektor występował efekt *pile-up*, czyli nakładanie się sygnałów.

Porównano punkty pomiarowe z dopasowaniem, a następnie zbadano wydajność detekcji w funkcji liczby zliczeń na sekundę (patrz Rysunek 10).



Rysunek 10: Wydajność detekcji w funkcji liczby zliczeń na sekundę

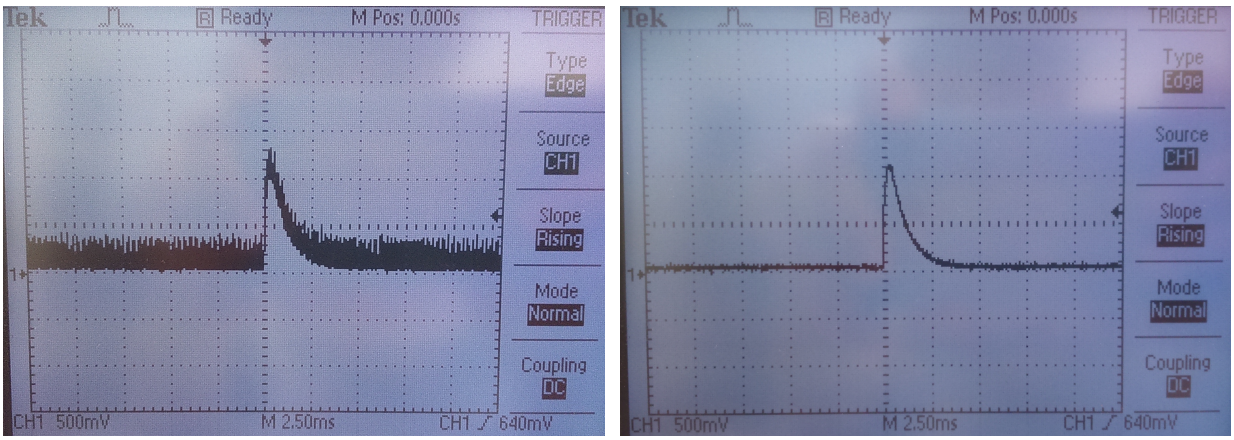
Do punktów dopasowano funkcję:

$$y(x) = \frac{100}{1 + (exp(\frac{x-a_0}{a_1}))^{a_2}}, \tag{2}$$

gdzie $a_0 = 5,09$, $a_1 = 0,17$ cm i $a_2 = 0,01$ cm są parametrami dopasowania.

5 Dyskusja i wnioski

Parametry wyznaczone podczas kalibracji zostały zaimplementowane w kodzie, co umożliwiło uwzględnienie wydajności prowadzonej detekcji. W czasie testowania skonstruowanego licznika odnotowano problemy ze zbyt wysokim poziomem szumu (patrz Rysunek 11), który utrudniał identyfikację sygnału pochodzącego z tuby G-M. Ustalono, że przyczyną było niewłaściwe uziemienie układu.



Rysunek 11: Sygnały na oscyloskopie odpowiednio: przed redukcją zaszumienia i po redukcji

Zauważono, iż maksymalna wartość napięcia generowanego przez przetwornicę silnie zależy od liczby i rodzaju połączonych z nią elementów. Przy braku obciążenia odbierane napięcie (mierzone na rozwartych zaciskach przetwornicy) sięga 440 V, natomiast po zmontowaniu układu spada ono do wartości ok. 330 V. Wartość ta wykracza poza zakres 360-440 V [3], zalecany dla użytego modelu tuby G-M.

Obecnie amplituda sygnału odbieranego na wejściu analogowym Arduino, generowanego przez tubę G-M przy wykryciu cząstki jonizującej, wynosi 1 V (patrz Rysunek 11). Nie wykorzystano zatem w pełni zakresu pomiarowego Arduino, który jest w stanie mierzyć napięcia do 5 V.

6 Podsumowanie

W ramach zespołowego projektu studenckiego skonstruowano poprawnie działający licznik Geigera-Müllera, który został skalibrowany pod względem wydajności detekcji. Licznik wyświetla całkowitą liczbę zliczeń, jakie zostały zarejestrowane od momentu uruchomienia detektora, oraz liczbę zliczeń na sekundę wyznaczoną z uwzględnieniem zależności wydajności detekcji od mierzonego poziomu promieniowania jonizującego.

6.1 Uwagi do dalszej pracy

Konstrukcja zbudowanego licznika umożliwia jego dalszą rozbudowę, dzięki czemu może on stać się tematem projektu zespołowego kolejnej grupy studentów zainteresowanych ochroną radiologiczną, chcących nabrać doświadczenia w pracy z prostymi układami elektronicznymi. Poniżej przedstawiono listę uwag i propozycji do dalszej pracy.

- Przy rozbudowie licznika G-M należy zwrócić uwagę na problemy z uziemieniem układu, mające znaczny wpływ na poziom szumów odbieranego sygnału.
- Do rozwiązania pozostaje problem niestabilności wysokiego napięcia generowanego przez przetwornicę.
- Obecną wersję detektora można zmodyfikować poprzez:
  - zmianę sposobu zasilania (akumulator, panel słoneczny),
  - umożliwienie komunikacji z detektorem przez łącze WiFi,
  - dodanie opcji wyświetlania dawki skutecznej, mocy dawki, po wykonaniu odpowiedniej kalibracji,
  - opracowanie sposobu wzmocnienia napięcia odbieranego na wejściu analogowym Arduino.

Podziękowania

Szczególnie dziękujemy Panu dr. hab. Krzysztofowi Miernikowi za poświęcony czas i kierowanie projektem, dzięki czemu plan konstrukcji licznika został pomyślnie zrealizowany. Serdeczne podziękowania kierujemy również do Pana Adama Kubieli za wsparcie techniczne oraz cenne wskazówki dotyczące lutowania i montowania układu.

7 Literatura

[1] K. S. Krane. *Introductory nuclear physics*. John Wiley & Sons (1988)

[2] S. Monk. *Arduino i Android. Niesamowite projekty*. Helion S.A. (2014)

[3] https://dozymetria.wordpress.com/2012/11/11/licznik-geigera-sts-5-gm-tube-ctc-5/