



---

# POLITECHNIKA POZNAŃSKA

---

WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

Instytut Informatyki

Praca dyplomowa licencjacka

**SPRZĘTOWY GENERATOR LICZB LOSOWYCH  
WYKORZYSTUJĄCY RZUTY KOSTKĄ**

Julia Samp, 151775

Jakub Kędra, 151790

Wojciech Kot, 151879

Jakub Prusak, 151178

Promotor  
dr inż. Jędrzej Potoniec

POZNAŃ 2025

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Podstawy teoretyczne</b>	<b>2</b>
<b>3</b>	<b>Fragment teoretyczny</b>	<b>3</b>
3.1	Zalety dostępnych rozwiązań TRNG . . . . .	3
3.2	Wady dostępnych rozwiązań TRNG . . . . .	3
3.2.1	Wydajność i szybkość generowania liczb losowych . . . . .	3
3.2.2	Koszty implementacji . . . . .	3
3.2.3	Stabilność i jakość generowanych liczb losowych . . . . .	4
3.2.4	Skomplikowana kalibracja i konserwacja . . . . .	4
3.2.5	Złożoność integracji z istniejącymi systemami . . . . .	4
3.2.6	Ograniczona dostępność i skalowalność . . . . .	4
3.2.7	Zagadnienia związane z bezpieczeństwem . . . . .	4
3.2.8	Podsumowanie wad TRNG . . . . .	4
3.3	Podsumowanie dostępnych na rynku TRNG . . . . .	5
<b>4</b>	<b>Przegląd komercyjnych rozwiązań TRNG</b>	<b>6</b>
4.1	Rozwiązania sprzętowe . . . . .	6
4.2	Generatory TRNG w chmurze . . . . .	6
4.3	Wady dostępnych rozwiązań TRNG . . . . .	7
4.3.1	Wydajność i szybkość generowania liczb losowych . . . . .	7
4.3.2	Koszty implementacji . . . . .	7
4.3.3	Stabilność i jakość generowanych liczb losowych . . . . .	7
4.3.4	Skomplikowana kalibracja i konserwacja . . . . .	8
4.3.5	Złożoność integracji z istniejącymi systemami . . . . .	8
4.3.6	Ograniczona dostępność i skalowalność . . . . .	8
4.3.7	Zagadnienia związane z bezpieczeństwem . . . . .	8
4.3.8	Podsumowanie wad TRNG . . . . .	8
4.4	Podsumowanie dostępnych na rynku TRNG . . . . .	8
<b>5</b>	<b>Rozwinięcie/test</b>	<b>9</b>
<b>6</b>	<b>Budowa sprzętowego generatora liczb losowych</b>	<b>10</b>
6.1	Projektowanie robota . . . . .	10
6.1.1	Prototypowanie . . . . .	10
6.1.2	Ostateczna wersja robota . . . . .	14
6.2	Dokumentacja techniczna . . . . .	17

6.2.1	Hardware . . . . .	17
Ogólne parametry robota: . . . . .	17	
Komponenty: . . . . .	17	
Diagramy i schematy: . . . . .	18	
6.2.2	Software . . . . .	18
<b>7</b>	<b>Testy</b>	<b>19</b>
7.1	Użyte wzory . . . . .	19
7.1.1	Test chi-kwadrat . . . . .	19
7.1.2	Entropia . . . . .	20
7.1.3	Test pojedynczych bitów . . . . .	20
7.1.4	Test serii . . . . .	20
7.1.5	Test najdłuższej serii . . . . .	20
7.1.6	Test pokerowy . . . . .	20
7.2	Testy poszczególnych wariantów . . . . .	21
7.2.1	Wariant 1 - „Betoniarka“ . . . . .	21
7.2.2	Wariant 2 - „Blender“ . . . . .	21
<b>8</b>	<b>Zakończenie</b>	<b>22</b>
<b>Literatura</b>		<b>23</b>
<b>A</b>	<b>Składanie dokumentu w systemie L<sup>A</sup>T<sub>E</sub>X</b>	<b>24</b>
A.1	Struktura dokumentu . . . . .	24
A.2	Akapity i znaki specjalne . . . . .	24
A.3	Wypunktowania . . . . .	24
A.4	Polecenia pakietu ppfcmtthesis . . . . .	25
A.5	Rysunki . . . . .	25
A.5.1	Tablice . . . . .	26
A.5.2	Checklista . . . . .	26
A.6	Literatura i materiały dodatkowe . . . . .	27

## Rozdział 1

### Wstęp

Wstęp do pracy powinien zawierać następujące elementy:

- krótkie uzasadnienie podjęcia tematu;
- cel pracy (patrz niżej);
- zakres (przedmiotowy, podmiotowy, czasowy) wyjaśniający, w jakim rozmiarze praca będzie realizowana;
- ewentualne hipotezy, które autor zamierza sprawdzić lub udowodnić;
- krótką charakterystykę źródeł, zwłaszcza literaturowych;
- układ pracy (patrz niżej), czyli zwięzłą charakterystykę zawartości poszczególnych rozdziałów;
- ewentualne uwagi dotyczące realizacji tematu pracy np. trudności, które pojawiły się w trakcie realizacji poszczególnych zadań, uwagi dotyczące wykorzystywanego sprzętu, współpraca z firmami zewnętrznymi.

No więc co do tego i tamtego testy kurde ten. lol.

xD

tak.

## Rozdział 2

# Podstawy teoretyczne

Rozdział teoretyczny — przegląd literatury naświetlający stan wiedzy na dany temat.

Przegląd literatury naświetlający stan wiedzy na dany temat obejmuje rozdziały pisane na podstawie literatury, której wykaz zamieszczany jest w części pracy pt. *Literatura* (lub inaczej *Bibliografia, Piśmiennictwo*). W tekście pracy muszą wystąpić odwołania do wszystkich pozycji zamieszczonych w wykazie literatury. **Nie należy odnośników do literatury umieszczać w stopce strony.** Student jest bezwzględnie zobowiązany do wskazywania źródeł pochodzenia informacji przedstawianych w pracy, dotyczy to również rysunków, tabel, fragmentów kodu źródłowego programów itd. Należy także podać adresy stron internetowych w przypadku źródeł pochodzących z Internetu.

## Rozdział 3

# Fragment teoretyczny

Współczesne systemy kryptograficzne oraz aplikacje wymagają generowania liczb losowych, które muszą charakteryzować się wysoką jakością i odpornością na przewidywalność. Najlepiej spełniają do generatory liczb losowych oparte na fizycznych zjawiskach, zwane TRNG (True Random Number Generator). TRNG są szczególnie istotne w kontekście aplikacji wymagających silnej ochrony danych, takich jak systemy kryptograficzne, podpisy cyfrowe, oraz w generowaniu kluczy szyfrujących.

### 3.1 Zalety dostępnych rozwiązań TRNG

AAAAAAAAAAAAA, AAAAAAAA AAA AAAAAAAA

### 3.2 Wady dostępnych rozwiązań TRNG

Mimo że Generatory Liczb Losowych Oparte na Zjawiskach Fizycznych (TRNG) oferują wysoki poziom bezpieczeństwa i losowości, istnieje kilka istotnych wad i ograniczeń związanych z ich implementacją i użytkowaniem. Wśród głównych problemów, które mogą wpływać na wydajność oraz niezawodność TRNG, wyróżnia się następujące aspekty:

#### 3.2.1 Wydajność i szybkość generowania liczb losowych

Wydajność TRNG jest często niższa niż w przypadku Generujących Liczby Losowe Oparte na Algorytmach (PRNG). Generowanie liczb losowych za pomocą zjawisk fizycznych, takich jak szum termiczny czy fluktuacje kwantowe, może być procesem czasochłonnym, szczególnie w systemach wymagających dużych ilości losowych liczb w krótkim czasie. W wyniku tego, TRNG mogą nie spełniać wymagań wydajnościowych w aplikacjach o dużym zapotrzebowaniu na losowość, takich jak systemy kryptograficzne o bardzo wysokiej częstotliwości operacji.

#### 3.2.2 Koszty implementacji

Urządzenia TRNG, zwłaszcza te oparte na zaawansowanych technologiach, takich jak fotonika kwantowa czy detekcja szumów kwantowych, mogą wiązać się z wysokimi kosztami produkcji oraz utrzymania. Z tego powodu, TRNG są często droższe w porównaniu do bardziej ekonomicznych rozwiązań opartych na algorytmach deterministycznych (PRNG), które wystarczają do wielu zastosowań, gdzie wysoka jakość losowości nie jest kluczowym wymaganiem.

### 3.2.3 Stabilność i jakość generowanych liczb losowych

Choć TRNG są uznawane za bezpieczne, ich jakość może być wpływana przez różne czynniki zewnętrzne, takie jak temperatura, zakłócenia elektromagnetyczne czy inne zmiany środowiskowe. W wyniku tego, generowane liczby losowe mogą wykazywać pewne niskiej jakości właściwości, co wymaga zastosowania dodatkowych mechanizmów, takich jak procesy post-przetwarzania, aby zapewnić ich odpowiednią losowość. Nawet małe zakłócenia w systemie mogą prowadzić do wzorców, które mogą zostać wykorzystane w atakach kryptograficznych.

### 3.2.4 Skomplikowana kalibracja i konserwacja

Urządzenia TRNG, szczególnie te, które wykorzystują skomplikowane zjawiska fizyczne, wymagają starannej kalibracji i ciągłego monitorowania, aby zapewnić ich prawidłowe funkcjonowanie. W wielu przypadkach konieczne jest stosowanie systemów nadzoru, które monitorują jakość generowanych liczb losowych w czasie rzeczywistym. Ponadto, wymogi dotyczące utrzymania odpowiednich warunków pracy, takich jak stabilna temperatura czy brak zakłóceń elektromagnetycznych, mogą stanowić dodatkową przeszkodę w ich szerokim zastosowaniu.

### 3.2.5 Złożoność integracji z istniejącymi systemami

Integracja TRNG z już działającymi systemami, zwłaszcza w kontekście urządzeń wbudowanych lub systemów o dużych wymaganiach obliczeniowych, może wiązać się z wieloma trudnościami. Często konieczne jest dostosowanie sprzętu lub oprogramowania w celu zapewnienia kompatybilności i pełnej funkcjonalności. Dodatkowo, ze względu na fizyczną naturę tych urządzeń, integracja z innymi komponentami może prowadzić do wzrostu kosztów oraz złożoności całego systemu.

### 3.2.6 Ograniczona dostępność i skalowalność

Choć rynek TRNG rośnie, nadal jest on stosunkowo niszowy w porównaniu do bardziej powszechnych rozwiązań opartych na PRNG. Ograniczona dostępność wyspecjalizowanych urządzeń TRNG, szczególnie tych, które oferują wysoką jakość generowanych liczb losowych, sprawia, że ich wdrożenie jest trudniejsze, zwłaszcza w przypadkach wymagających masowej produkcji. Ponadto, nie wszystkie rozwiązania są skalowalne, co może być problemem w przypadku aplikacji wymagających elastyczności i łatwego dostosowywania wydajności do rosnących potrzeb.

### 3.2.7 Zagadnienia związane z bezpieczeństwem

Chociaż TRNG zapewniają wyższy poziom bezpieczeństwa niż PRNG, nie są one całkowicie odporne na ataki. Ataki fizyczne, takie jak manipulacje w obrębie urządzenia lub przechwytywanie sygnałów z jego elementów, mogą prowadzić do kompromitacji jakości liczb losowych. Ponadto, w przypadku rozwiązań opartych na technologii chmurowej, takich jak te oferowane przez Cloudflare, istnieje ryzyko ataków związanych z przechwytywaniem lub manipulowaniem danymi w trakcie transmisji, co może wpływać na bezpieczeństwo generowanych liczb losowych.

### 3.2.8 Podsumowanie wad TRNG

Chociaż TRNG oferują niezrównaną jakość losowości w porównaniu do rozwiązań opartych na algorytmach, posiadają również szereg wad, które mogą ograniczać ich szerokie zastosowanie.

Należy do nich niska wydajność, wysokie koszty implementacji, problemy ze stabilnością generowanych liczb losowych, złożoność integracji z systemami oraz ryzyko związane z bezpieczeństwem. W miarę jak technologia będzie się rozwijać, możliwe jest, że te ograniczenia zostaną przezwyciężone, jednak obecnie stanowią one istotne wyzwanie dla szerokiego przyjęcia TRNG w różnych aplikacjach.

### 3.3 Podsumowanie dostępnych na rynku TRNG

## Rozdział 4

# Przegląd komercyjnych rozwiązań TRNG

Współczesne systemy kryptograficzne oraz aplikacje wymagają generowania liczb losowych, które muszą charakteryzować się wysoką jakością i odpornością na przewidywalność. Najlepiej spełniają do generatory liczb losowych oparte na fizycznych zjawiskach, zwane TRNG (True Random Number Generator). TRNG są szczególnie istotne w kontekście aplikacji wymagających silnej ochrony danych, takich jak systemy kryptograficzne, podpisy cyfrowe, oraz w generowaniu kluczy szyfrujących.

### 4.1 Rozwiązania sprzętowe

Wśród komercyjnych rozwiązań sprzętowych, wiodącymi producentami są firmy takie jak **ID Quantique**, **Microchip Technology** i **Semtech**, które oferują zaawansowane urządzenia bazujące na TRNG. Produkty te zapewniają wysoki poziom bezpieczeństwa i są stosowane w wymagających aplikacjach, takich jak bankowość elektroniczna czy systemy wojskowe.

- **ID Quantique** jest jednym z liderów w dziedzinie generatorów liczb losowych opartych na technologii fotoniki. Firma oferuje urządzenia, które wykorzystują detekcję fotonów w celu generowania liczb losowych. Dzięki temu rozwiązania ID Quantique charakteryzują się bardzo wysoką jakością losowości, a jednocześnie są odporne na ataki związane z analizą i przewidywaniem generowanych liczb.
- **Microchip Technology** w swojej ofercie posiada różne moduły TRNG, w tym układy scalone, które generują liczby losowe na podstawie fluktuacji szumów termicznych. Produkty te znajdują zastosowanie w szerokim zakresie aplikacji, od urządzeń mobilnych po systemy wbudowane.
- **Semtech** natomiast oferuje rozwiązania, które wykorzystują zjawiska losowe zachodzące w obwodach analogowych do generowania liczb losowych. Firma ta jest jednym z głównych dostawców układów TRNG, które znajdują szerokie zastosowanie w urządzeniach IoT oraz w systemach komunikacji bezprzewodowej.

### 4.2 Generatory TRNG w chmurze

W ostatnich latach pojawiły się także rozwiązania chmurowe, które umożliwiają generowanie liczb losowych w czasie rzeczywistym bez potrzeby posiadania własnego sprzętu. Przykładem takiego rozwiązania jest **Cloudflare** – firma specjalizująca się w dostarczaniu usług związanych z

bezpieczeństwem sieciowym. Na swoim blogu Cloudflare przedstawia zaawansowane metody generowania liczb losowych, które są wykorzystywane w ich systemach. Cloudflare korzysta z technologii opartych na zjawiskach fizycznych, takich jak detekcja fizycznych "bąbli" w lampach lawowych ("Entropy Wall"), nieprzewidywalnego fizycznie trójstopniowego wahadła, rozpadu radioaktywnego Uranu oraz procesów związane z tzw. "chaosem kwantowym". Tego typu rozwiązania pozwalają na szybkie i bezpieczne generowanie liczb losowych w skali globalnej, zapewniając jednocześnie wysoki poziom ochrony przed atakami.

Firma ta oferuje użytkownikom dostęp do generatora liczb losowych w chmurze, który jest wykorzystywany m.in. do tworzenia kluczy kryptograficznych oraz w innych zastosowaniach wymagających silnych zabezpieczeń. Dzięki wykorzystaniu globalnej infrastruktury Cloudflare, generowane liczby losowe są szeroko dostępne i charakteryzują się dużą niezawodnością oraz odpornością na ataki.

### 4.3 Wady dostępnych rozwiązań TRNG

Mimo że Generatory Liczb Losowych Oparte na Zjawiskach Fizycznych (TRNG) oferują wysoki poziom bezpieczeństwa i losowości, istnieje kilka istotnych wad i ograniczeń związanych z ich implementacją i użytkowaniem. Wśród głównych problemów, które mogą wpływać na wydajność oraz niezawodność TRNG, wyróżnia się następujące aspekty:

#### 4.3.1 Wydajność i szybkość generowania liczb losowych

Wydajność TRNG jest często niższa niż w przypadku Generujących Liczby Losowe Oparte na Algorytmach (PRNG). Generowanie liczb losowych za pomocą zjawisk fizycznych, takich jak szum termiczny czy fluktuacje kwantowe, może być procesem czasochłonnym, szczególnie w systemach wymagających dużych ilości losowych liczb w krótkim czasie. W wyniku tego, TRNG mogą nie spełniać wymagań wydajnościowych w aplikacjach o dużym zapotrzebowaniu na losowość, takich jak systemy kryptograficzne o bardzo wysokiej częstotliwości operacji.

#### 4.3.2 Koszty implementacji

Urządzenia TRNG, zwłaszcza te oparte na zaawansowanych technologiach, takich jak fotonika kwantowa czy detekcja szumów kwantowych, mogą wiązać się z wysokimi kosztami produkcji oraz utrzymania. Z tego powodu, TRNG są często droższe w porównaniu do bardziej ekonomicznych rozwiązań opartych na algorytmach deterministycznych (PRNG), które wystarczają do wielu zastosowań, gdzie wysoka jakość losowości nie jest kluczowym wymaganiem.

#### 4.3.3 Stabilność i jakość generowanych liczb losowych

Choć TRNG są uznawane za bezpieczne, ich jakość może być wpływana przez różne czynniki zewnętrzne, takie jak temperatura, zakłócenia elektromagnetyczne czy inne zmiany środowiskowe. W wyniku tego, generowane liczby losowe mogą wykazywać pewne niskiej jakości właściwości, co wymaga zastosowania dodatkowych mechanizmów, takich jak procesy post-przetwarzania, aby zapewnić ich odpowiednią losowość. Nawet małe zakłócenia w systemie mogą prowadzić do wzorców, które mogą zostać wykorzystane w atakach kryptograficznych.

#### 4.3.4 Skomplikowana kalibracja i konserwacja

Urządzenia TRNG, szczególnie te, które wykorzystują skomplikowane zjawiska fizyczne, wymagają starannej kalibracji i ciągłego monitorowania, aby zapewnić ich prawidłowe funkcjonowanie. W wielu przypadkach konieczne jest stosowanie systemów nadzoru, które monitorują jakość generowanych liczb losowych w czasie rzeczywistym. Ponadto, wymogi dotyczące utrzymania odpowiednich warunków pracy, takich jak stabilna temperatura czy brak zakłóceń elektromagnetycznych, mogą stanowić dodatkową przeszkodę w ich szerokim zastosowaniu.

#### 4.3.5 Złożoność integracji z istniejącymi systemami

Integracja TRNG z już działającymi systemami, zwłaszcza w kontekście urządzeń wbudowanych lub systemów o dużych wymaganiach obliczeniowych, może wiązać się z wieloma trudnościami. Często konieczne jest dostosowanie sprzętu lub oprogramowania w celu zapewnienia kompatybilności i pełnej funkcjonalności. Dodatkowo, ze względu na fizyczną naturę tych urządzeń, integracja z innymi komponentami może prowadzić do wzrostu kosztów oraz złożoności całego systemu.

#### 4.3.6 Ograniczona dostępność i skalowalność

Choć rynek TRNG rośnie, nadal jest on stosunkowo niszowy w porównaniu do bardziej powszechnych rozwiązań opartych na PRNG. Ograniczona dostępność wyspecjalizowanych urządzeń TRNG, szczególnie tych, które oferują wysoką jakość generowanych liczb losowych, sprawia, że ich wdrożenie jest trudniejsze, zwłaszcza w przypadkach wymagających masowej produkcji. Ponadto, nie wszystkie rozwiązania są skalowalne, co może być problemem w przypadku aplikacji wymagających elastyczności i łatwego dostosowywania wydajności do rosnących potrzeb.

#### 4.3.7 Zagadnienia związane z bezpieczeństwem

Chociaż TRNG zapewniają wyższy poziom bezpieczeństwa niż PRNG, nie są one całkowicie odporne na ataki. Ataki fizyczne, takie jak manipulacje w obrębie urządzenia lub przechwytywanie sygnałów z jego elementów, mogą prowadzić do kompromitacji jakości liczb losowych. Ponadto, w przypadku rozwiązań opartych na technologii chmurowej, takich jak te oferowane przez Cloudflare, istnieje ryzyko ataków związanych z przechwytywaniem lub manipulowaniem danymi w trakcie transmisji, co może wpływać na bezpieczeństwo generowanych liczb losowych.

#### 4.3.8 Podsumowanie wad TRNG

Chociaż TRNG oferują niezrównaną jakość losowości w porównaniu do rozwiązań opartych na algorytmach, posiadają również szereg wad, które mogą ograniczać ich szerokie zastosowanie. Należy do nich niska wydajność, wysokie koszty implementacji, problemy ze stabilnością generowanych liczb losowych, złożoność integracji z systemami oraz ryzyko związane z bezpieczeństwem. W miarę jak technologia będzie się rozwijać, możliwe jest, że te ograniczenia zostaną przezwyciężone, jednak obecnie stanowią one istotne wyzwanie dla szerokiego przyjęcia TRNG w różnych aplikacjach.

### 4.4 Podsumowanie dostępnych na rynku TRNG

## Rozdział 5

# Rozwinięcie/test

Rozdziały dokumentujące pracę własną studenta: opisujące ideę, sposób lub metodę rozwiązania postawionego problemu oraz rozdziały opisujące techniczną stronę rozwiązania — dokumentacja techniczna, przeprowadzone testy, badania i uzyskane wyniki.

Praca musi zawierać elementy pracy własnej autora adekwatne do jego wiedzy praktycznej uzyskanej w okresie studiów. Za pracę własną autora można uznać np.: stworzenie aplikacji informatycznej lub jej fragmentu, zaproponowanie algorytmu rozwiązania problemu szczegółowego, przedstawienie projektu np. systemu informatycznego lub sieci komputerowej, analizę i ocenę nowych technologii lub rozwiązań informatycznych wykorzystywanych w przedsiębiorstwach, itp.

Autor powinien zadbać o właściwą dokumentację pracy własnej obejmującą specyfikację założeń i sposób realizacji poszczególnych zadań wraz z ich oceną i opisem napotkanych problemów. W przypadku prac o charakterze projektowo-implementacyjnym, ta część pracy jest zastępowana dokumentacją techniczną i użytkową systemu.

W pracy **nie należy zamieszować całego kodu źródłowego** opracowanych programów. Kod źródłowy napisanych programów, wszelkie oprogramowanie wytworzone i wykorzystane w pracy, wyniki przeprowadzonych eksperymentów powinny być umieszczone np. na płycie CD, stanowiącej dodatek do pracy.

## Styl tekstu

Należy<sup>1</sup> stosować formę bezosobową, tj. *w pracy rozważono ....., w ramach pracy zaprojektowano .....*, a nie: *w pracy rozważyłem, w ramach pracy zaprojektowałem*. Odwołania do wcześniejszych fragmentów tekstu powinny mieć następującą postać: „Jak wspomniano wcześniej, ....”, „Jak wykazano powyżej ....”. Należy unikać długich zdań.

Niedopuszczalne są zwroty używane w języku potocznym. W pracy należy używać terminologii informatycznej, która ma sprecyzowaną treść i znaczenie.

Niedopuszczalne jest pisanie pracy metodą *cut&paste*, bo jest to plagiat i dowód intelektualnej indolencji autora. Dane zagadnienie należy opisać własnymi słowami. Zawsze trzeba powołać się na zewnętrzne źródła.

---

<sup>1</sup>Uwagi o stylu pochodzą częściowo ze stron prof. Macieja Drozdowskiego [1].

## Rozdział 6

# Budowa sprzętowego generatora liczb losowych

### 6.1 Projektowanie robota

#### 6.1.1 Prototypowanie

Przy całym procesie budowy robota wykorzystano technologię druku 3D. Pozwala na szybkie modyfikacje przy jednoczesnym zachowaniu bardzo wysokiej dokładności i jakości wykonania elementów. Dzięki temu podczas budowy prototypów a później ostatecznej wersji robota można było w prosty sposób modyfikować, projektować i zmieniać każdy element składowy robota. Pomimo tego, że proces druku 3D w szczególności dużych skomplikowanych elementów jest dość czasochłonny to jest to najlepsza dostępna nam technologia do wykorzystania w tego rodzaju projektach.

Głównym założeniem podczas projektowania i budowy robota było założenie modułowości. Oznacza to, że każdy element jest wymienny i łatwo dostępny. Dzięki takiemu podejściu wymienianie elementów w przypadku awarii czy też małe modyfikacje wynikające z udoskonalania działania robota są znaczco prostsze i przede wszystkim szybsze niż gdyby cały robot był jednolitą bryłą.

Proces projektowania robota rozpoczęto od przeanalizowania różnych metod wykonywania rzutu kością. Po przeanalizowaniu kilku pomysłów, zdecydowano się na rozwiązanie wykorzystujące obrotowy kubek, wewnątrz którego kość porusza się i odbija od ścianek. Wariant ten roboczo nazwano betoniarką od podobnej zasady działania. Taki mechanizm zapewnia rzut zbliżony do takiego wykonanego przez człowieka, a jednocześnie jest prosty w konstrukcji. Obracający się kubek został zaprojektowany tak, aby jego prędkość obrotową i czas trwania obrotu można było w prosty sposób kontrolować. W celach testowych został skonstruowany prototypowy model robota.

Pierwszy prototyp robota składał się z metalowych prętów służących za stelaż oraz elementów wydrukowanych na drukarce 3D. Tymi elementami były: kubek, ramie służące do montażu kubka, uchwyty do prętów oraz płytka mocująca do kamery. Dodatkowo wykorzystano silnik prądu stałego napędzający kubek oraz sterownik służący do zasilania i sterowania ruchem silnika.



RYSUNEK 6.1: bałagan na stole

Po pierwszych testach okazało się, że niezbędny do uzyskania zamierzonego efektu będzie mechanizm, który będzie wychylał cały kubek wraz z silnikiem, który odpowiada za jego obrót. Z początku planowano wykorzystanie serwomechanizmu jednak to rozwiązanie odrzucono, ponieważ większość dostępnych serwomechanizmów, ma ograniczony obrót do  $180^\circ$  lub  $360^\circ$  a to limitowały możliwości mechanizmu służącego do wychylania kubka. Ostatecznie w tym celu wybrano mały silnik krokowy z wystarczającym do wychylenia kubka momentem obrotowym ( $34.3\text{mN.m}$ ). Silnik ten obraca układem dwóch kół zębatych.



RYSUNEK 6.2: zebatki

Do pierwszych testów robota zaprojektowano kilka wariantów kształtów kubków. Najlepszym wariantem okazał się być cylindryczny kubek z dodatkowymi pionowymi żebrami, o które kostka się odbijała podczas kręcenia.

Podczas pierwszej wersji robota wykorzytującej obrotowy kubek powstał pomysł alternatywnego rozwiązania. Rozwiązanie to implementuje inne podejście do rzutu kością. Zamiast obracać cały kubek, a dodatkowo wyhylać go, wykorzystany został trwale zamontowany kubek, na którego dnie znajduje się śmieci, które podcina leżącą na dnie kością. Takie rozwiązanie znacząco

upraszcza cały mechanizm robota oraz bardzo przyspiesza proces losowania liczby. Ten wariant nazwano roboczo blenderem również od podobnej zasady działania.

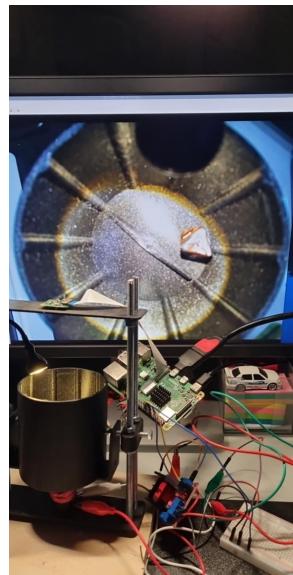
Przy projektowaniu drugiego wariantu robota został wykorzystany ten sam stelaż złożony z metalowych prętów co w pierwszym wariantie. Na drukarce 3D wydrukowano dodatkowe części, niezbędne do realizacji tego wariantu. Zaprojektowano i wydrukowano nowy kubek, śmiegle oraz mocowanie dla silnika. Kubek został przystosowany do montażu silnika prądu stałego oraz śmieglą.

W trakcie testów zauważono, że procesor robota nagrzewa się do wysokich temperatur podczas intensywnej pracy, co mogło negatywnie wpływać na jego wydajność i żywotność. Aby temu zapobiec, w projekcie zdecydowano się na zastosowanie radiatorów, które miały pomóc w rozproszeniu nadmiaru ciepła, oraz wentylatora, który wspomagał cyrkulację powietrza wokół procesora. Dzięki temu rozwiązaniu udało się obniżyć temperaturę pracy procesora, która teraz utrzymuje się w granicach 40°C, co zapewnia stabilne i bezpieczne działanie całego systemu.

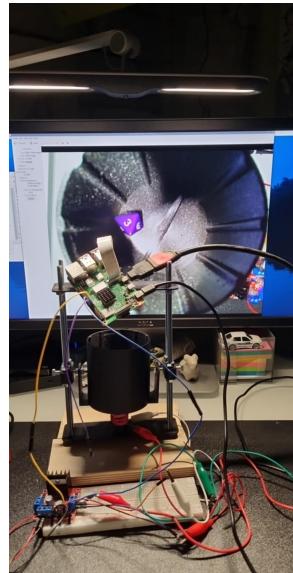


RYSUNEK 6.3: zimno

Duże znaczenie ma również wykorzystywana kość. Od jej koloru i tekstury zależy jakość zdjęć zrobionych przez zamonotowaną kamerę. Poniżej przedstawiono dwa przykłady zdjęć i różnic w ich czytelności zależnych od koloru kości.



RYSUNEK 6.4: gorzej



RYSUNEK 6.5: lepiej

W obu wariantach dużym problemem był słaby obraz z kamery. W tym celu zaprojektowano system oświetlenia składający się z diod LED sterowanych za pomocą układu ULN2803A Darlington. Dzięki temu wnętrze kubka stało się dużo jaśniejsze co pozwala kamerze na robienie zdjęć lepszej jakości. Dodatkową rozświetlenie wnętrza kubka na tyle poprawiło jakość zdjęć, że pozwoliło to na obniżenie kamery względem kubka. Spowodowało to że wysokość prototypu zmniejszyła się o około 5cm - co również miało duże znaczenie podczas projektowania robota, ponieważ jednym z założeń postawionych na początku budowy była kompaktowość. Dzięki tym zabiegom otrzymywane zdjęcia stały się dużo bardziej wyraźne oraz pole widzenia kamery było ograniczone do dna kubka. Diody LED służące do oświetlenia wnętrza kubka połączono szeregowo dzięki czemu nie trzeba było wykorzystywać dodatkowych rezystorów a ilość połączeń została minimalna.



RYSUNEK 6.6: jasno

Po skonstruowaniu obu wersji robota oczywistym stało się, że wariant blendera będzie znacznie szybciej (około 3-krotnie) generował wyniki. Dodatkowo był ona zdecydowanie prostszy w budowie przez mniejszą liczbę ruchomych elementów. Z tego powodu do docelowego robota wybrano wariant blendera. Dzięki temu projekt stał się mniej skomplikowany mechanicznie a jednocześnie jego użyteczność wzrosła, ponieważ głównym zadaniem tego robota jest generowanie liczb losowych a to zadanie szybciej był w stanie wykonywać wariant blendera.

### 6.1.2 Ostateczna wersja robota

Projektowanie ostatecznej wersji robota rozpoczęto od przeanalizowania wad, zalet oraz ogólnych cech (takich jak wymiary) prototypów. Postanowiono, że finalna wersja robota będzie wykorzystywała kubek o tej samej średnicy co prototypowa. Dookoła tych wymiarów zaprojektowano resztę konstrukcji. Obudowę zaprojektowano w taki sposób żeby była w stanie pomieścić kubek oraz kamerę umieszczoną na wysokości 120mm nad dnem kubka. Wysokość tą określono eksperymentalnie podczas testów prototypów. Dodatkowo w tylnej oraz dolnej części obudowy pozustawiono przestrzeń na resztę elementów składowych robota. Tą przestrzeń wyznaczono poprzez zwymiarowanie pozostałych elementów takich jak silnik, sterownik, wentylator czy Raspberry Pi. Te wymiary posłużyły do wyznaczenia minimum potrzebnej przestrzeni. Do tego minimum dodano (CO) w taki sposób żeby we wnętrzu pomieściły się również przewody i śruby oraz żeby po całkowitym złożeniu pozostała przestrzeń do swobodnego manipulowania elementami składowymi.

Każdy element został zaprojektowany w taki sposób żeby cały robot był modułowy. Żeby spełnić to założenie, każdy element zaprojektowano w taki sposób żeby posiadał on specjalne miejsca na inserty lub śruby. Inserty to mosiężne elementy, które za pomocą lutownicy wgrzewają się w wydruk 3D. Posiadają one wewnętrzny gwint dzięki czemu można dołączenia wydruków wykorzystywać śruby nie uszkadzając samego wydruku przy wielokrotnym skręcaniu i rozkręcaniu robota.

Górna część obudowy - pokrywa - mieszcząca kamerę oraz diody LED, została zaprojektowana w taki sposób żeby dostęp do kubka pozostał łatwy. Osiągnięto to poprzez wykorzystanie prostego mocowania pokrywy do obudowy, z wykorzystaniem pojedyńczej śruby oraz magnesów neodynowych. Dzięki temu w przypadku kiedy konieczny jest dostęp do kamery lub diod LED wystarczy zdjąć tylną ścianę robota oraz odkręcić pojedyńczą śrubę. Dodatkowo magnesy neodymowe zapewniają dobre przyleganie pokrywy do reszty obudowy. Ich dodatkową zaletą jest blokowanie pokrywy w ustalonej pozycji podczas umieszczania kości wewnętrz kubka.

Kamera wraz z diodami LED służącymi do oświetlenia wnętrza kubka znajduje się w pokrywie. Kamere zamontowano przy użyciu dwóch śrub M2 natomiast diody LED umieszczone w zaprojektowanych w wydruku otworach.

W pokrywie znajdują się również dwa sześciokątne otwory na magnesy neodymowe. Dzięki temu że otwory są sześciokątne to walcowe magnesy idealnie się w nie wpasowują. W drugiej części obudowy znajdują się takie same otwory na drugą parę magnesów. Dla pewności podczas umieszczania magnesów wykorzystano klej cyjanoakrylowy.

Kubek, w którym dokonywane są rzuty kością, zaprojektowano na podstawie kubka z prototypowej wersji blendera. Zachowano jego średnicę oraz kształt wewnętrznych żeber. Zmieniony został natomiast sposób mocowania kubka w taki sposób żeby był on przystosowany do zamocowania w obudowie. W tym celu zaprojektowano cztery mocowania za pomocą, których kubek jest przykręcany do obudowy. Dodatkowo pogrubiono dno kubka tak żeby można było w nim zamocować inserty służące do mocowania uchwytu silnika. Ostatnią modyfikacją było podwyższenie kubka w taki sposób żeby wysokość sięgała on aż do mocowania kamery - górnej pokrywy. Dzięki temu podczas rzutów zniknął problem z wypadającą kośćią co było dość częstym zajwiskiem podczas testów prototypu. Niestety takie Rozwiązywanie spowodowało, że wewnętrzne kubka przestało być widoczne z zewnątrz. Jednak zostało uznane, że nie jest to konieczne do osiągnięcia celów projektu.

Projekt pierwszego uchwytu mocującego silnik składał się z miejsca do umieszczenia silnika oraz dwóch cylindrycznych słupków służących za prowadnice śrub mocujących cały uchwyt do dna kubka. Jednak, ponieważ podczas testów pojawiły się problemy z pierwszym silnikiem i wymienionego na inny konieczne było zaprojektowanie drugiego uchwytu. Podczas projektowania tego drugiego uchwytu wzorowano się na projekcie pierwszego jednak dodano dodatkowy trzeci cylindryczny słupek na śrubę, która miała służyć do kontrolowania wychylenia całego uchwytu na boki.

Do mocowania Raspberry Pi zaprojektowano specjalną płytę mocowaną do boku obudowy. W projekcie tej płytki uwzględniono otwory do przymocowania Raspberry Pi oraz układu ULN2803A Darlington. Dodatkowo przygotowano specjalne miejsce do mocowania przycisku. Na płytce pozostało również miejsce na zamontowanie szyny zasilania. Płytkę tą umieszczono w obudowie w taki sposób żeby znajdowała się ona bezpośrednio nad wentylatorem. Dzięki temu strumień powietrza bezpośrednio chłodzi najważniejsze elementy elektroniczne robota. Przycisk zamocowano na tej samej płytce z wykorzystaniem dodatkowego elementu wydrukowanego na drukarce 3D. Dzięki temu znalazł się on bezpośrednio przy ściance obudowy a dodatkowo jego mocowanie nadal spełnia założenie modułowości poprzez mocowanie za pomocą śrub i insertów.

Układ sterujący silnikiem LN298 przykręcono do obudowy pośrednio poprzez specjalnie zaprojektowane mocowanie. Dzięki temu wykorzystano gotowe otwory na śruby znajdujące się w płytce układu LN298.

Do dna obudowy robota przymocowano za pomocą śrub również wentylator.

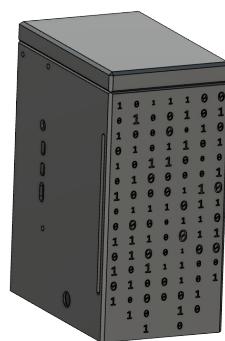
Obudowę zaprojektowano w taki sposób żeby pomieściła wszystkie powyższe elementy. W jej scianach zaprojektowano otwory na wyjścia Raspberry Pi, diody LED oraz gniazdo zasilania. W scianie obudowy na wysokości miejsca, w którym znajduje się wewnętrzny przycisk, zaprojektowano specjalne wycięcie. Dzięki dodatkowemu zmniejszeniu grubości ściany obudowy w tym miejscu jest bardziej elastyczna co pozwala na kliknięcie przycisku znajdującego się po wewnętrznej stronie ściany obudowy. Na dnie obudowy zaprojektowano również specjalne słupki służące za podpórki dla kubka. W słupkach tych zaprojektowano otwory na inserty dzięki, którym kubek można przykręcić do obudowy gwarantując tym jego stabilność.

Podczas projektowania obudowy przewidziano także takie elementy jak wycięcia od spodu bezpośrednio pod wentylatorem służące za wlot powietrza oraz wycięcia w tylnej ściance służące za wyłot powietrza. Dodatkowo w dnie umieszczono duży utwór możliwiający swobodne mocowanie oraz dostęp do silnika, diod LED oraz gniazda zasilania.

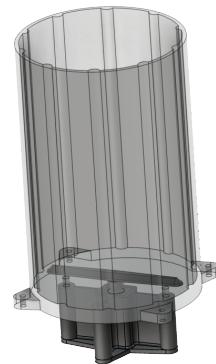
Podczas testów ostatecznej wersji robota napotkano problem z pierwotnie wykorzystywanym silnikiem. Silnik ten miał okrągły wał przez co śmigło musiało być bardzo dokładnie spasowane

aby uniknąć ślizgania się wału wewnątrz otworu śmiega. To sprawiało, że montowanie śmiega i jego demontaż był bardzo trudny. Dodatkowo po wielokrotnych rzutach kością, śmigło wbijało się coraz niżej na wał silnika i w ostateczności tarło o dno kubka tak mocno, że silnik nie był w stanie się obracać. Żeby temu zaradzić umieszczono pomiędzy śmigłem a dnem kubka metalową podkładkę po której śmigło mogło się ślizgać łatwiej niż po dnie kubka. To jednak nie rozwiązało problemu ponieważ po kolejnych kilku tysiącach rzutów śmigło zaczęło się blokować. Z tego powodu postanowiono wymienić silnik na mocniejszy 12V silnik z przekładnią, który ma mniejszą częstotliwość obrotu (około 1000RPM zamiast 4000RPM) ale ma większy moment obrotowy. Dodatkową zaletą nowego silnika jest jego wał w kształcie litery "D". Pozwala to na znacznie luźniejsze spasowanie otworu śmiega z wałem przez co demontaż śmigła jest znacznie łatwiejszy.

Zdjecia:



RYSUNEK 6.7: gotowy



RYSUNEK 6.8: kubek



RYSUNEK 6.9: Płytyka do montażu elektroniki

## 6.2 Dokumentacja techniczna

### 6.2.1 Hardware

#### Ogólne parametry robota:

Wymiary:

Waga:

#### Komponenty:

Elektronika:

- Raspberry Pi 4B,
- Raspberry Pi Camera V2,
- LN298,
- ULN2803A Darlington,
- silnik DC 12V,
- diody LED 3mm,
- przycisk monostabilny THT.

Elementy strukturalne (Wydrukowane na drukarce 3D. Materiał, którego użyto to PLA):

- obudowa,
- tylna ściana obudowy,
- górna pokrywa,
- płytka do montażu kamery i diod LED,
- uchwyt do silnika,
- płytka do montażu elektroniki,

- śmigło.

Zasilanie: Zasilacz DC 12V.

**Diagramy i schematy:**

### **6.2.2 Software**

bbbbbbbbbbbbbbbbbbbb

## Rozdział 7

# Testy

Przez wzgląd na brak oprogramowania, nie da się przeprowadzić typowej oceny jakości projektu. Zamiast tego, postanowiono sprawdzić losowość odczytanych wyników rzutu kością dla obu wariantów robota, a następnie porównać je między sobą oraz z losością wbudowanego generatora pseudolosowego programu Python.

Do oceny losowości otrzymanych wyników użyto:

- test chi-kwadrat,
- test pojedynczych bitów,
- test serii,
- test długiej serii,
- test pokerowy
- procentową wartość entropii.

Dla otrzymania wystarczającej ilości danych, należało wygenerować 20 000 bitów. Ponieważ robot używa kości ośmiościennnej, dającą wynik w postaci dokładnie 3 bitów, należało wykonać 6 667 rzutów.

### 7.1 Użyte wzory

#### 7.1.1 Test chi-kwadrat

Aby przeprowadzić ten test, dla wszystkich  $n$  rzutów zliczono, ile razy wypadła każda z  $k$  ścian kości ośmiościennej ( $O_i$ ). Wyniki te porównano z oczekiwana liczbą wyrzucenia każdej ze ścianek  $E_i = \frac{n}{k}$ .

$$\chi^2 = \sum_{i=1}^k \left( \frac{O_i - E_i}{E_i} \right)^2$$

Ponieważ do generowania liczb losowych użyto kości ośmiościennej, to  $k$  jest równe 8, co daje wzór:

$$\chi^2 = \sum_{i=1}^8 \left( \frac{O_i - E_i}{E_i} \right)^2$$

Dla przeprowadzonych testów wybrano poziom istotności  $p = 0,5$ , natomiast stopień swobody przy  $k = 8$  równa się:

$$k - 1 = 8 - 1 = 7$$

Zatem dla otrzymanego stopnia swobody oraz założonego stopnia istotności, wartość krytyczna wynosi 14,0671. Test jest zakończony sukcesem, jeśli otrzymana wartość będzie mniejsza od wartości krytycznej.

### 7.1.2 Entropia

$$H(X) = - \sum_{i=1}^k p(x_i) \log_2 p(x_i)$$

Ponieważ wartość entropii jest wyznaczana dla bitów, a każdy rzut na kości ośmiościennej daje wynik w 3 bitach, wartość entropii przeprowadzonego testu może wynosić maksymalnie 3.

### 7.1.3 Test pojedynczych bitów

Dla wszystkich wygenerowanych bitów zliczono, ile razy wystąpiły 0 ( $n_0$ ) i ile razy wystąpiły 1 ( $n_1$ ). Test jest zakończony sukcesem, jeśli każda z tych wartości spełnia zależność:

$$9725 < n_0 < 10275$$

$$9725 < n_1 < 10275$$

### 7.1.4 Test serii

Jednocześnie dla ciągu wygenerowanych bitów zliczono tzw. serie, czyli ile razy bez przerw występuje podciąg zer lub jedynek o długościach od jednego do pięciu takiego samego bitu pod rząd. Dłuższe serie zliczono do jednej kategorii. Aby test został zakończony sukcesem, liczba serii danych długości dla zarówno zer jak i jedynek musi spełniać zależność przedstawioną w poniższej tabeli.

Długość serii	Przedział
1	2315 - 2685
2	1114 - 1386
3	527 - 723
4	240 - 384
5	103 - 209
6 i więcej	103 - 209

### 7.1.5 Test najdłuższej serii

Przy zliczaniu bitów dla testu serii, jednocześnie szukano najdłuższego podciągu samych zer i najdłuższego podciągu samych jedynek. Test długiej serii jest zakończony sukcesem, jeśli żaden z tych podciągów nie zawiera więcej niż 25 bitów.

### 7.1.6 Test pokerowy

Do przeprowadzenia testu pokerowego podzielono wygenerowany ciąg na czwórki bitów, a następnie zliczono, ile razy wystąpiła każda z szesnastu kombinacji  $s_i$ . Test jest zakończony sukcesem, jeśli spełniona jest poniższa nierówność:

$$2,16 < X < 46,17$$

Gdzie:

$$X = \frac{16}{5000} \sum_{i=0}^{15} s_i^2 - 5000$$

## 7.2 Testy poszczególnych wariantów

Coś tam coś tam, mamy dwa warianty robota to możemy je porównać. (jeśli będzie czas na testowanie betoniarki)

### 7.2.1 Wariant 1 - „Betoniarzka“

### 7.2.2 Wariant 2 - „Blender“

## Rozdział 8

### Zakończenie

Zakończenie pracy zwane również Uwagami końcowymi lub Podsumowaniem powinno zawierać ustosunkowanie się autora do zadań wskazanych we wstępie do pracy, a w szczególności do celu i zakresu pracy oraz porównanie ich z faktycznymi wynikami pracy. Podejście takie umożliwia jasne określenie stopnia realizacji założonych celów oraz zwrócenie uwagi na wyniki osiągnięte przez autora w ramach jego samodzielnej pracy.

Integralną częścią pracy są również dodatki, aneksy i załączniki zawierające stworzone w ramach pracy programy, aplikacje i projekty.

# Literatura

- [1] Maciej Drozdowski. Jak pisać prace dyplomowe – uwagi o formie. [on-line]  
[http://www.cs.put.poznan.pl/mdrozdowski/dyd/txt/jak\\_mgr.html](http://www.cs.put.poznan.pl/mdrozdowski/dyd/txt/jak_mgr.html), 2006.
- [2] Donald E. Knuth. *The TeXbook*. Computers and Typesetting. Addison-Wesley, Reading, MA, USA, 1986.
- [3] Leslie Lamport. *LaTeX — A Document Preparation System — User’s Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.

## Dodatek A

# Składanie dokumentu w systemie $\text{\LaTeX}$

W tym rozdziale znajduje się garść informacji o tym, jak poprawnie składać tekst pracy w systemie  $\text{\LaTeX}$  wraz z przykładami, które mają służyć do przeklejania do własnych dokumentów.

### A.1 Struktura dokumentu

Praca składa się z rozdziałów (`chapter`) i podrozdziałów (`section`). Ewentualnie można również rozdziały zagnieżdzać (`subsection`, `subsubsection`), jednak nie powinno się wykraczać poza drugi poziom hierarchii (czyli `subsubsection`).

### A.2 Akapity i znaki specjalne

Akapity rozdziela się od siebie przynajmniej jedną pustą linią. Podstawowe instrukcje, które się przydają to *wyróżnienie pewnych słów*. Można również stosować **styl pogrubiony**, choć nie jest to generalnie zalecane.

Należy pamiętać o zasadach polskiej interpunkcji i ortografii. Po spójnikach jednoliterowych warto wstawić znak tyldy (~), który jest tak zwaną „twardą spacją” i powoduje, że wyrazy nią połączone nie będą rozdzielane na dwie linie tekstu.

Polskie znaki interpunkcyjne różnią się nieco od angielskich: to jest „polski”, a to jest “angielski”. W kodzie źródłowym tego tekstu będzie widać różnicę.

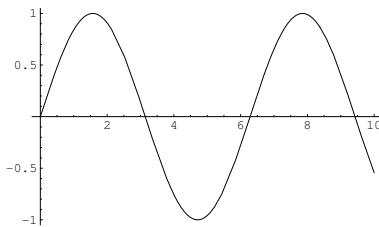
Proszę również zwrócić uwagę na znak myślnika, który może być pauzą „—” lub półpauzą: „–”. Należy stosować je konsekwentnie. Do łączenia wyrazów używamy zwykłego „-” (*północnowschodni*), do myślników — pauzy lub półpauzy. Inne zasady interpunkcji i typografii można znaleźć w słownikach.

### A.3 Wypunktowania

Wypunktowanie z cyframi:

1. to jest punkt,
2. i to jest punkt,
3. a to jest ostatni punkt.

Po wypunktowaniach czasem nie warto wstawiać wcięcia akapitowego. Wtedy przydatne jest polecenie `noindent`. Wypunktowanie z kropkami (tzw. *bullet list*) wygląda tak:



RYSUNEK A.1: Wykres.

- to jest punkt,
- i to jest punkt,
- a to jest ostatni punkt.

Wypunktowania opisowe właściwie niewiele się różnią:

**elementA** to jest opis,

**elementB** i to jest opis,

**elementC** a to jest ostatni opis.

## A.4 Polecenia pakietu *ppfcmtthesis*

Parę poleceń zostało zdefiniowanych aby uspójnić styl pracy. Są one przedstawione poniżej (oczywiście nie trzeba się do nich stosować).

**Makra zdefiniowane dla języka angielskiego.** Są nimi: `termdef` oraz `acronym`. Przykłady poniżej obrazują ich przewidywane użycie w tekście.

---

źródło	<code>we call this a \termdef{Database Management System} (\acronym{DBMS})</code>
docelowo	<i>we call this a Database Management System (DBMS)</i>

---

**Makra zdefiniowane dla języka polskiego.** Podobnie jak dla języka angielskiego zdefiniowano odpowiedniki polskie: `definicja`, `akronim` oraz `english` dla tłumaczeń angielskich terminów. Przykłady poniżej obrazują ich przewidywane użycie w tekście.

---

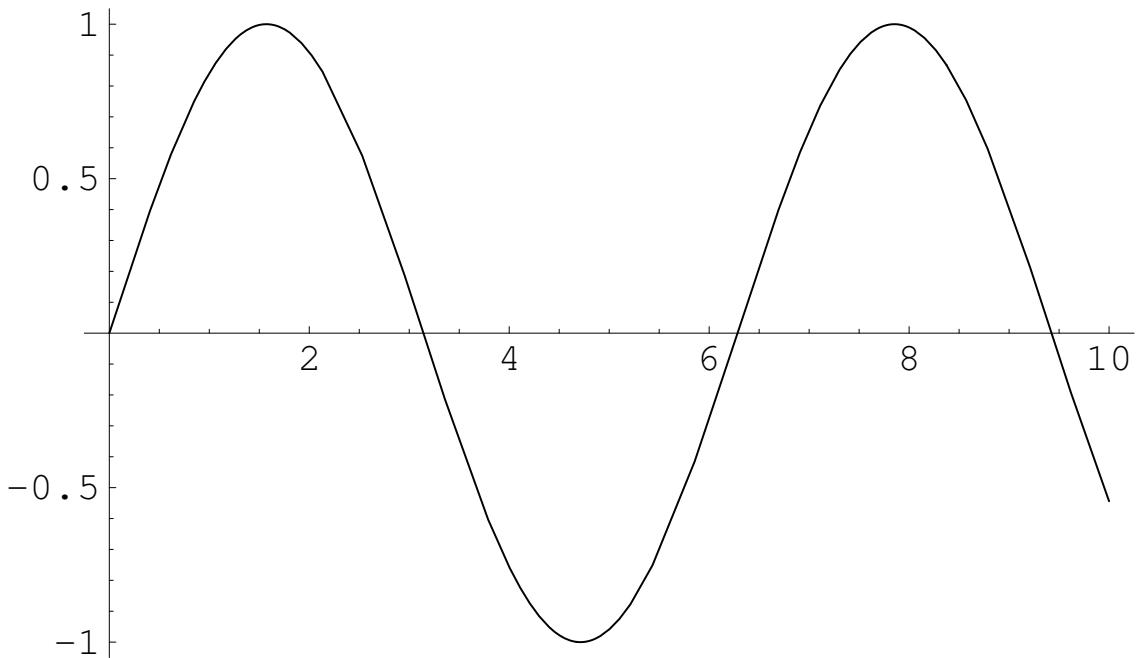
źródło	<code>nazywamy go \definicja{systemem zarządzania bazą danych} (\akronim{DBMS}, \english{Database Management System})</code>
docelowo	<i>nazywamy go systemem zarządzania bazą danych (DBMS, ang. Database Management System)</i>

---

## A.5 Rysunki

Wszystkie rysunki (w tym również diagramy, szkice i inne) osadzamy w środowisku `figure` i umieszczamy podpis *pod* rysunkiem, w formie elementu `caption`. Rysunki powinny zostać umieszczone u góry strony (osadzone bezpośrednio w treści strony zwykle utrudniają czytanie tekstu). Rysunek A.1 zawiera przykład pełnego osadzenia rysunku na stronie.

Styl FCMu to nieco inne nagłówki rysunków. Dostępne są one poleceniem `fcmfcaption` (zob. rysunek A.2).



**Rysunek A.2.** Ten sam wykres ale na szerokość tekstu. Formatowanie podpisu zgodne z wytycznymi FCMu.

### A.5.1 Tablice

Tablice to piękna rzecz, choć akurat ich umiejętne tworzenie w  $\text{\LaTeX}x$  nie jest łatwe. Jeśli tablica jest skomplikowana, to można ją na przykład wykonać w programie OpenOffice, a następnie wyeksportować jako plik *PDF*. W każdym przypadku tablice wstawia się podobnie jak rysunki, tylko że w środowisku `table`. Tradycja typograficzna sugeruje umieszczenie opisu tablicy, a więc elementu `caption` ponad jej treścią (inaczej niż przy rysunkach).

Tablica A.1 pokazuje pełen przykład.

TABELA A.1: Przykładowa tabela. Styl opisu jest zgodny z rysunkami.

artykuł	cena [zł]
bułka	0,4
masło	2,5

Zasady FCMu sugerują nieco inne nagłówki tablic. Dostępne są one poleceniem `fcmtcaption` (zob. tablicę A.2).

**Tabela A.2**  
Przykładowa tabela. Styl opisu jest zgodny z wytycznymi FCMu.

artykuł	cena [zł]
bułka	0,4
masło	2,5

### A.5.2 Checklista

- Znakiem myślnika jest w  $\text{\LaTeX}x$  dywiz pełen (—) albo półpauza (–), przykład: A niech to jasna cholera — wrzasnąłem.

- Połączenie między wyrazami to zwykły myślnik, przykład: północno-zachodni
- Sprawdź czy tytuł pracy ma maksymalnie dwa wiersze i czy stanowią one pełne frazy (czy nie ma przeniesienia bez sensu).
- Sprawdź ostrzeżenia o 'overfull' i 'underful' boxes. Niektóre z nich można zignorować (spójrz na wynik formatowania), niektóre trzeba poprawić; czasem przeformułować zdanie.
- item Przypisy stawia się wewnątrz zdań lub za kropką, przykład: Footnote is added after a comma.<sup>1</sup>
- Nie używaj przypisów zbyt często. Zobacz, czy nie lepiej będzie zintegrować przypis z tekstem.
- Tytuły tabel, rysunków powinny kończyć się kropką.
- Nie używaj modyfikatora [h] (here) do rysunków i tabel. Rysunki i tabele powinny być justowane do góry strony lub na stronie osobnej.
- Wyróżnienie w tekście to polecenie *wyraz*, nie należy używać czcionki pogrubionej (która wystaje wizualnie z tekstu i rozprasza).
- Nazwy plików, katalogów, ścieżek, zmiennych środowiskowych, klas i metod formatujemy poleciением `plik_o_pewnej_nazwie`.
- Po ostatniej zmianie do treści, sprawdź i przenieś wiszące spójniki wstawiając przed nie znak tyldy (twardej spacji), przykład: Ala i kotek nie lubią mleczka, a Stasiu lubi.
- Za i.e. (id est) i e.g. (exempli gratia) stawia się zwyczajowo przecinek w typografii amerykańskiej.
- Przed i za pełną pauzą nie ma zwyczajowo spacji w typografii amerykańskiej, przykład: Darn, this looks good—said Mary.
- Zamkający cudzysłów oraz footnote wychodzą za ostatni znak interpunkcji w typografii amerykańskiej, przykłady: It can be called a “curiosity,” but it’s actually normal. Footnote is added after a comma.<sup>2</sup>
- Odwołania do tabel i rysunków zawsze z wielkiej litery, przykład: In Figure A.1 we illustrated XXX and in Table A.1 we show detailed data.

## A.6 Literatura i materiały dodatkowe

Materiałów jest mnóstwo. Oto parę z nich:

- *The Not So Short Introduction...*, która posiada również tłumaczenie w języku polskim.  
<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>
- Klasy stylu `memoir` posiadają bardzo wiele informacji o składzie tekstów anglosaskich oraz sposoby dostosowania L<sup>A</sup>T<sub>E</sub>Xa do własnych potrzeb.  
<http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf>
- Nasza grupa dyskusyjna i repozytorium Git są również dobrym miejscem aby zapytać (lub sprawdzić czy pytanie nie zostało już zadane).  
<https://github.com/politechnika/put-latex>

---

<sup>1</sup>Here is a footnote.

<sup>2</sup>Here is a footnote.

- Dla łaknących więcej wiedzy o systemie LaTeX podstawowym źródłem informacji jest książka Lamporta [3]. Prawdziwy *hardcore* to oczywiście *The T<sub>E</sub>Xbook* profesora Knutha [2].



© 2025 Julia Samp, Jakub Kędra, Wojciech Kot, Jakub Prusak

Instytut Informatyki, Wydział Informatyki i Telekomunikacji  
Politechnika Poznańska

Skład przy użyciu systemu L<sup>A</sup>T<sub>E</sub>X - MiK<sup>T</sup>eX oraz workflow za pomocą Github Actions.