

Informatyka 1 (ES1E2009)

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr II, studia stacjonarne I stopnia
Rok akademicki 2019/2020

Wykład nr 1 (25.02.2020)

dr inż. Jarosław Forenc

Dane podstawowe

- dr inż. Jarosław Forenc
- Politechnika Białostocka, Wydział Elektryczny,
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki
ul. Wiejska 45D, 15-351 Białystok
WE-204
- e-mail: j.forenc@pb.edu.pl ■ tel. (0-85) 746-93-97
- <http://jforenc.prv.pl>
 - Dydaktyka - dodatkowe materiały do pracowni specjalistycznej
- konsultacje:
 - poniedziałek, godz. 09:00-10:00, WE-204
 - wtorek, godz. 11:00-12:00, WE-204
 - środa, godz. 09:00-10:00 + 12:15-13:00, WE-204
 - piątek, godz. 17:00-18:30, WE-204 (studia zaoczne)

Program wykładu (1/2)

1. Informacja analogowa i cyfrowa. Pozycyjne i niepozycyjne systemy liczbowe. Konwersje pomiędzy systemami liczbowymi.
2. Jednostki informacji cyfrowej. Kodowanie informacji. Kodowanie znaków.
3. Kodowanie liczb. Reprezentacja liczb w systemach komputerowych: stałoprzecinkowa i zmiennoprzecinkowa. Standard IEEE 754.
4. Programowanie w języku C. Deklaracje i typy zmiennych, operatory i wyrażenia arytmetyczne, operacje wejścia-wyjścia, operatory relacyjne i logiczne, wyrażenia logiczne, instrukcja warunkowa if, instrukcja switch, operator warunkowy, pętle (for, while, do .. while), tablice jednowymiarowe.

Program wykładu (2/2)

5. Architektura komputerów. Klasyfikacja systemów komputerowych (taksonomia Flynna). Architektura von Neumana i architektura harwardzka.
6. Budowa i zasada działania komputera. Procesor, pamięć wewnętrzna i zewnętrzna. Komunikacja z urządzeniami zewnętrznymi, interfejsy komputerowe.
7. Algorytmy. Definicja algorytmu. Klasyfikacje i sposoby przedstawiania algorytmów. Rekurencja. Złożoność obliczeniowa. Sortowanie. Klasyfikacje algorytmów sortowania. Wybrane algorytmy sortowania.
8. Zaliczenie wykładu.

Literatura (1/2)

1. R. Kawa, J. Lembas: „Wykłady z informatyki. Wstęp do informatyki”. PWN, Warszawa, 2017.
2. W. Kwiatkowski: „Wprowadzenie do kodowania”. BEL Studio, Warszawa, 2010.
3. S. Gryś: „Arytmetyka komputerów w praktyce”. PWN, Warszawa, 2013.
4. W. Stallings: „Organizacja i architektura systemu komputerowego. Projektowanie systemu a jego wydajność”. WNT, Warszawa, 2004.
5. A.S. Tanenbaum: „Strukturalna organizacja systemów komputerowych”. Helion, Gliwice, 2006.
6. K. Wojtuszkiewicz: „Urządzenia techniki komputerowej. Część 1. Jak działa komputer? Część 2. Urządzenia peryferyjne i interfejsy”. PWN, Warszawa, 2011.

Literatura (2/2)

7. K. Banasiak: „Algorytmizacja i programowanie w Matlabie”. Wydawnictwo BTC, Legionowo, 2017.
8. P. Wróblewski: „Algorytmy, struktury danych i techniki programowania. Wydanie V”. Helion, Gliwice, 2015.
9. M. Sysło: „Algorytmy”. Helion, Gliwice, 2016.
10. B. Buczek: „Algorytmy. Ćwiczenia”. Helion, Gliwice, 2008.
11. G. Coldwin: „Zrozumieć programowanie”. PWN, Warszawa, 2015.
12. S. Prata: „Język C. Szkoła programowania. Wydanie VI”. Helion, Gliwice, 2016.

Efekty uczenia się i system ich oceniania

Podstawę do zaliczenia przedmiotu (uzyskanie punktów ECTS) stanowi stwierdzenie, że **każdy** z założonych **efektów uczenia się** został osiągnięty w co najmniej minimalnym akceptowalnym stopniu.

EU1	identyfikuje i opisuje zasadę działania podstawowych elementów systemu komputerowego
EU2	formułuje algorytmy komputerowe rozwiązujące typowe zadania inżynierskie występujące w elektrotechnice

Zaliczenie wykładu - efekty uczenia się (EU1)

- Student, który zaliczył przedmiot:
 - identyfikuje i opisuje zasadę działania podstawowych elementów systemu komputerowego
- Student, który zalicza na ocenę **dostateczny (3)**:
 - wymienia podstawowe elementy systemu komputerowego i podaje ich przeznaczenie
 - krótko charakteryzuje klasyfikację Flynna systemów komputerowych
 - wyjaśnia podstawowe pojęcia związane z architekturą i zasadą działania systemów komputerowych
 - dokonuje konwersji liczby całkowitej bez znaku z systemu dziesiętnego na system o dowolnej podstawie i z systemu o dowolnej podstawie na system dziesiętny



Zaliczenie wykładu - efekty uczenia się (EU1)

- Student, który zalicza na ocenę **dostateczny (3)** (c.d.):
 - wyjaśnia na czym polega zapis zmiennoprzecinkowy liczby rzeczywistej oraz postać znormalizowana tego zapisu
- Student, który zalicza na ocenę **dobry (4)** (oprócz wymagań na ocenę 3):
 - opisuje strukturę i zasadę działania wybranych elementów systemu komputerowego
 - wymienia różnice pomiędzy architekturą von Neumana i architekturą harwardzką systemów komputerowych
 - dokonuje konwersji liczby całkowitej ze znakiem na wybrany kod (ZM, U1, U2) i odwrotnie
 - charakteryzuje wybrane kody liczbowe (NKB, BCD, Graya) i alfanumeryczne (ASCII, ISO-8859, Unicode)

Zaliczenie wykładu - efekty uczenia się (EU1)

- Student, który zalicza na ocenę **bardzo dobry (5)** (oprócz wymagań na ocenę 4):
 - przedstawia cel stosowania oraz zasadę działania pamięci podręcznej
 - omawia sposób kodowania wartości specjalnych w standardzie IEEE 754

Zaliczenie wykładu - efekty uczenia się (EU2)

- Student, który zaliczył przedmiot:
 - formułuje algorytmy komputerowe rozwiązujące typowe zadania inżynierskie występujące w elektrotechnice
- Student, który zalicza na ocenę **dostateczny (3)**:
 - przedstawia rozwiązanie prostego problemu w postaci schematu blokowego opisującego algorytm komputerowy
 - podaje definicję algorytmu komputerowego i wymienia metody opisu algorytmów
 - przedstawia sposób sortowania wektora liczb stosując wybraną, prostą metodę sortowania

Zaliczenie wykładu - efekty uczenia się (EU2)

- Student, który zalicza na ocenę **dobry (4)** (oprócz wymagań na ocenę 3):
 - przedstawia rozwiązanie złożonego problemu w postaci schematu blokowego opisującego algorytm komputerowy
 - wyjaśnia pojęcie złożoności obliczeniowej algorytmu, podaje złożoności obliczeniowe przykładowych algorytmów
- Student, który zalicza na ocenę **bardzo dobry (5)** (oprócz wymagań na ocenę 4):
 - wyjaśnia pojęcie rekurencji i podaje przykłady algorytmów rekurencyjnych
 - przedstawia sposób sortowania wektora liczb stosując metodę sortowania szybkiego (Quick-Sort)

Zaliczenie wykładu

- Sprawdzian pisemny:
 - sprawdzian: 16.06.2020 (wtorek), godz. 12:15, WE-Aula III
 - poprawa: termin do ustalenia
- Na zaliczeniu oceniane będą dwa efekt uczenia się (EU1, EU2)
- Za każdy efekt uczenia się można otrzymać od 0 do 100 pkt.
- Na podstawie otrzymanych punktów wystawiana jest ocena:

Punkty	Ocena	Punkty	Ocena
91 - 100	5,0	61 - 70	3,5
81 - 90	4,5	51 - 60	3,0
71 - 80	4,0	0 - 50	2,0

Terminy zajęć

- Wykład nr 1 - 25.02.2020
- Wykład nr 2 - 10.03.2020
- Wykład nr 3 - 24.03.2020
- Wykład nr 4 - 07.04.2020
- Wykład nr 5 - 05.05.2020
- Wykład nr 6 - 19.05.2020
- Wykład nr 7 - 02.06.2020
- Wykład nr 8 - 16.06.2020 (1 h, 12:15-13:00)

Zaliczenie wykładu

- Prowadzący zajęcia może przyznawać dodatkowe punkty za aktywność na wykładzie
- Ocena końcowa wyznaczana jest na podstawie sumy otrzymanych punktów:

Punkty	Ocena	Punkty	Ocena
182 - 200	5,0	122 - 141	3,5
162 - 181	4,5	102 - 121	3,0
142 - 161	4,0	0 - 101	2,0

Plan wykładu nr 1

- Język C
 - historia, struktura programu
 - kompilacja, zapis kodu
 - sekwencje sterujące, komentarze
- Pojęcia: informatyka i informacja
- Informacja analogowa i cyfrowa
- Systemy liczbowe
 - liczby i cyfry
 - systemy pozycyjne (dziesiętny, dwójkowy, szesnastkowy)
 - systemy niepozycyjne (rzymski)
- Konwersje między systemami liczbowymi

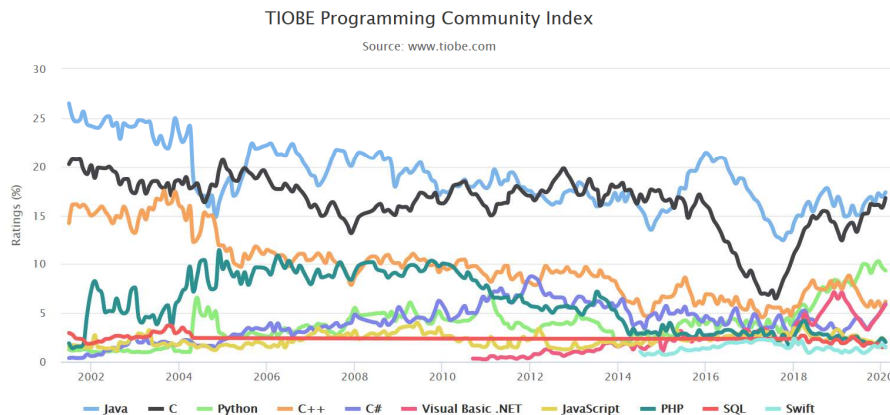
Język C - krótka historia (1/2)

- **1969** - język BCPL - Martin Richards, University Mathematical Laboratories, Cambridge
- **1970** - język B - Ken Thompson, adaptacja języka BCPL dla pierwszej instalacji systemu Unix na komputer DEC PDP-7
- **1972** - język NB (New B), nazwany później C - Dennis Ritchie, Bell Laboratories, New Jersey, system Unix na komputerze DEC PDP-11
 - 90% kodu systemu Unix oraz większość programów działających pod jego kontrolą napisane w C
- **1978** - książka „The C Programming Language” (Kernighan, Ritchie), pierwszy podręcznik, nieformalna definicja standardu (K&R)

Język C - krótka historia (2/2)

- **1989** - standard ANSI X3.159-1989 „Programming Language C” (ANSI C, C89)
- **1990** - adaptacja standardu ANSI C w postaci normy ISO/IEC 9899:1990 (C90)
- **1999** - norma ISO/IEC 9899:1999 (C99)
- **2011** - norma ISO/IEC 9899:2011 (C11)
- **2018** - norma ISO/IEC 9899:2018 (C18 lub C17)

Język C - TIOBE Programming Community Index



Język C - pierwszy program

- Niesformatowany plik tekstowy o odpowiedniej składni i mający rozszerzenie **.c**
- Kod najprostszego programu:

```
#include <stdio.h>

int main(void)
{
    printf("Witaj swiecie\n");
    return 0;
}
```

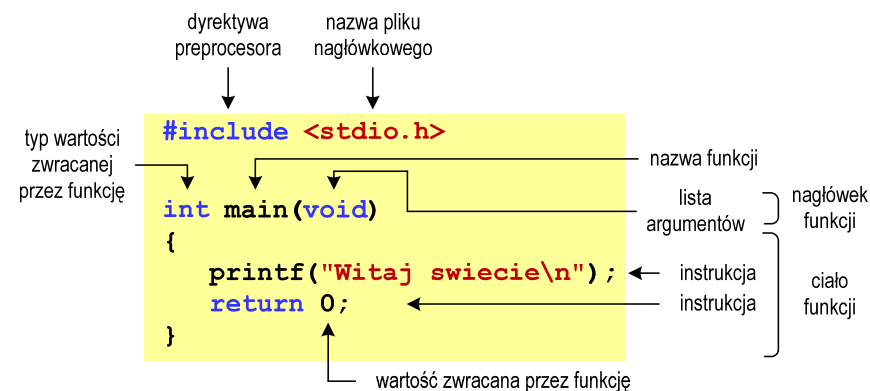
- Program konsolowy - wyświetla w konsoli tekst **Witaj swiecie**

Język C - pierwszy program

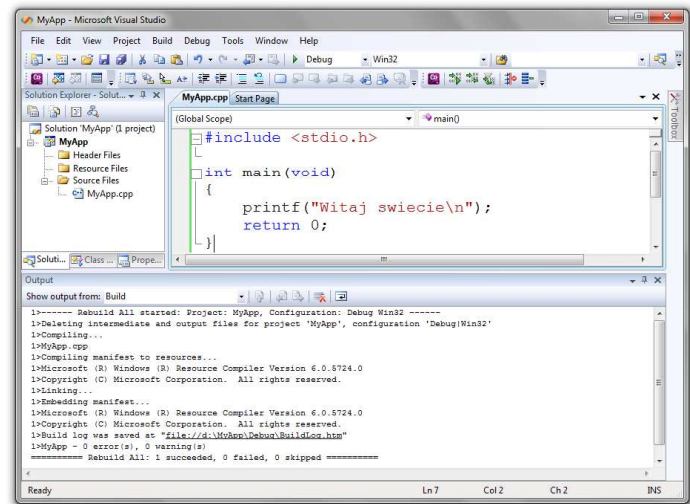
- Wynik uruchomienia programu:



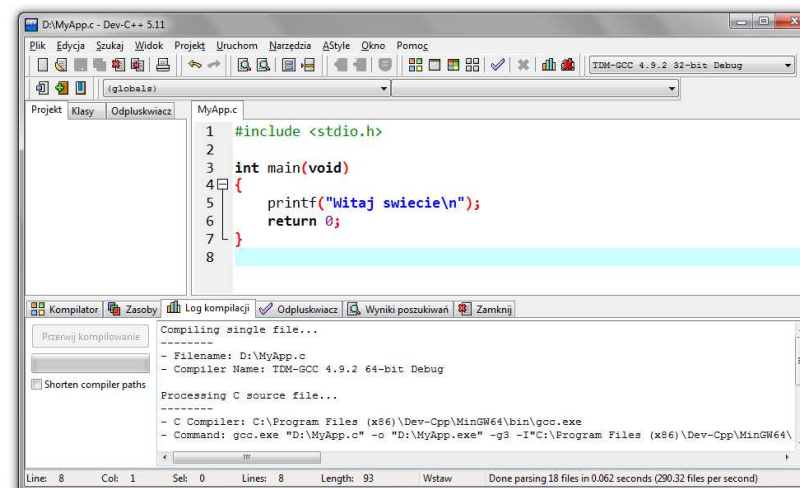
Język C - struktura programu



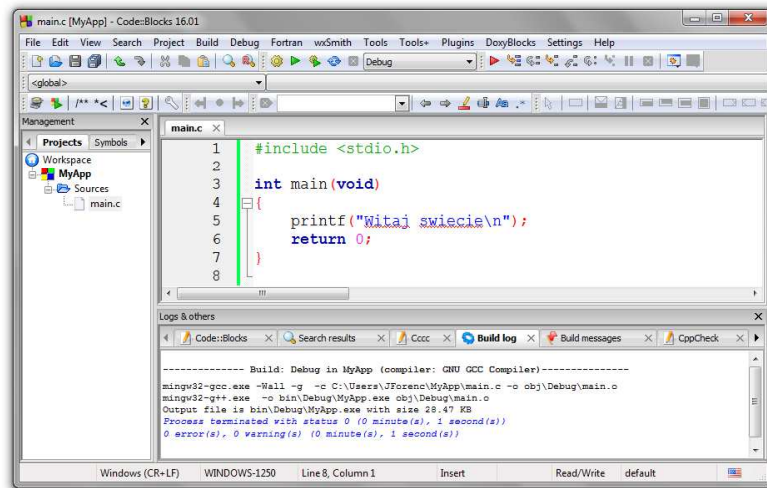
Microsoft Visual Studio 2008



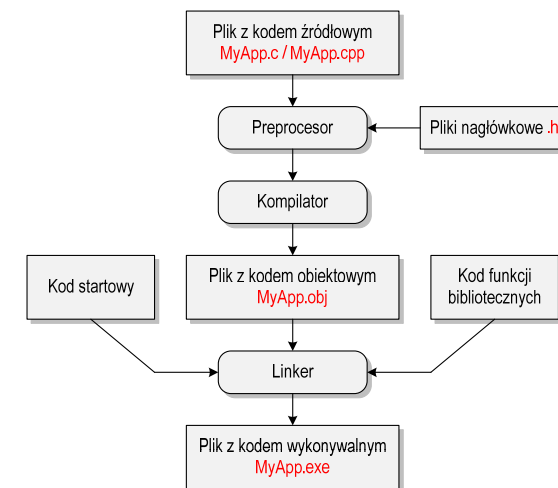
Dev-C++ 5.11



Code::Blocks 16.01



Język C - kompilacja programu



Język C - zapis kodu programu

- Sposób zapisu kodu programu wpływa tylko na jego przejrzystość, a nie na kompilację i wykonanie
- W takiej postaci program także skompiluje się:

```
#include <stdio.h>
int main(void) {printf("Witaj swiecie\n"); return 0;}
```

- W Microsoft Visual Studio 2008 można automatycznie sformatować kod źródłowy programu - **Ctrl + K + F**
- Język C rozróżnia **wielkość liter** - poniższy kod nie skompiluje się:

```
#include <stdio.h>
int Main(void) {printf("Witaj swiecie\n"); return 0;}
```

Język C - Wyświetlanie tekstu (printf)

- Znak przejścia do nowego wiersza `\n` może pojawić w dowolnym miejscu łańcucha znaków

```
printf("Witaj swiecie\n");
```

Witaj swiecie

—

```
printf("Witaj\nswiecie\n");
```

Witaj
swiecie

—

```
printf("Witaj ");
printf("swiecie");
printf("\n");
```

Witaj swiecie

—

Język C - Sekwencje sterujące

- Istnieją także inne sekwencje sterujące (ang. escape sequence)

Opis znaku	Zapis w printf()
Alarm (ang. alert), głośniczek wydaje dźwięk	<code>\a</code>
Backspace	<code>\b</code>
Wysunięcie strony (ang. form feed)	<code>\f</code>
Przejsięcie do nowego wiersza (ang. new line)	<code>\n</code>
CR - Carriage Return (powrót na początek wiersza)	<code>\r</code>
Tabulacja pozioma (odstęp) (ang. horizontal tab)	<code>\t</code>
Tabulacja pionowa (ang. vertical tab)	<code>\v</code>

Język C - Wyświetlenie znaków specjalnych

- Niektóre znaki pełnią specjalną funkcję i nie można wyświetlić ich w tradycyjny sposób

Opis znaku	Znak	Zapis w printf()
Cudzysłów	"	<code>\"</code>
Apostrof	'	<code>\'</code>
Ukośnik (ang. backslash)	\	<code>\\</code>
Procent	%	<code>%%</code>

Sciezka dostępu: "C:\dane\plik.txt"

```
printf("Sciezka dostepu: \\\"C:\\dane\\plik.txt\\\"\\n");
```

Język C - Wyświetlenie znaku o podanym kodzie

- Można wyświetlić dowolny znak podając jego kod w systemie ósemkowym lub szesnastkowym

Znaczenie	Zapis
Znak o podanym kodzie ASCII (system ósemkowy)	<code>\0oo</code>
Znak o podanym kodzie ASCII (system szesnastkowy)	<code>\xhh</code>

```
printf("\127\151\164\141\152\040");  
printf("\x73\x77\x69\x65\x63\x69\x65\x21\x0A");
```

Witaj swiecie!

Język C - Wyświetlenie tekstu

```
#include <stdio.h>  
  
int main(void)  
{  
    printf("-----\n");  
    printf("| Punkty | Ocena |\n");  
    printf("-----\n");  
    printf("| 91-100 | 5,0 |\n");  
    printf("| 81-90 | 4,5 |\n");  
    printf("| 71-80 | 4,0 |\n");  
    printf("| 61-70 | 3,5 |\n");  
    printf("| 51-60 | 3,0 |\n");  
    printf("| 0-50 | 2,0 |\n");  
    printf("-----\n");  
  
    return 0;  
}
```

Punkty	Ocena
91-100	5,0
81-90	4,5
71-80	4,0
61-70	3,5
51-60	3,0
0-50	2,0

Język C - Komentarze

- Komentarze są pomijane podczas kompilacji

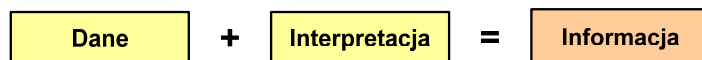
```
/*  
  Nazwa: MyApp.cpp  
  Autor: Jarosław Forenc, Politechnika Białostocka  
  Data: 25-02-2020 12:15  
  Opis: Program wyświetlający tekst "Witaj świecie"  
*/  
  
#include <stdio.h>    // zawiera deklarację printf()  
  
int main(void)        // nagłówek funkcji main()  
{  
    printf/* funkcja */("Witaj świecie\n");  
  
    return 0;  
}
```

Informatyka

- **Informatyka** (ang. computer science)
 - dziedzina nauki i techniki zajmująca się gromadzeniem, przetwarzaniem i wykorzystywaniem **informacji**
 - w języku polskim termin informatyka zaproponował w październiku 1968 r. prof. Romuald Marczyński na konferencji poświęconej „maszynom matematycznym”
 - wzorem nazwy były francuskie **informatique** i niemieckie **Informatik**
- **Informatykę** można rozpatrywać jako:
 - samodzielną dyscyplinę naukową
 - narzędzie wykorzystywane przez inne nauki
 - gałąź techniki
 - przemysł wytwarzający sprzęt (hardware) i oprogramowanie (software)

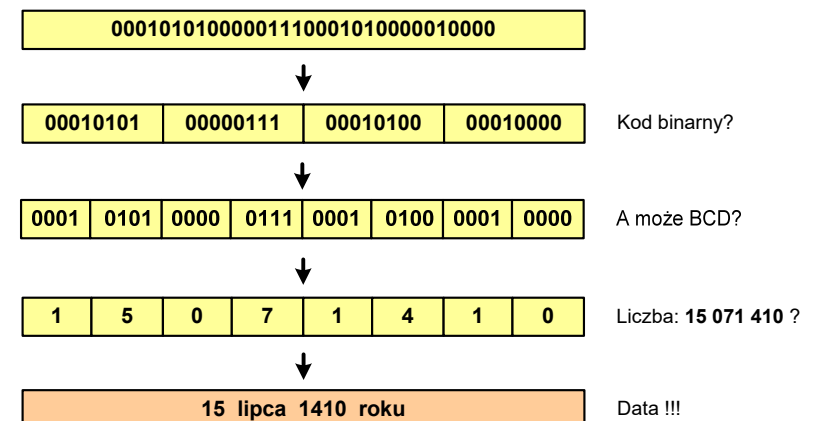
Informacja

- **Informatyka** (ang. computer science)
 - dziedzina nauki i techniki zajmująca się gromadzeniem, przetwarzaniem i wykorzystywaniem **informacji**
- **Informacja** - wielkość abstrakcyjna, która może być:
 - przechowywana w pewnych obiektach
 - przesyłana pomiędzy pewnymi obiektami
 - przetwarzana w pewnych obiektach
 - stosowana do sterowania pewnymi obiektami
- **Dane** - surowe fakty i liczby
- **Przetwarzanie danych** - logicznie powiązany zespół czynności pozwalających na uzyskanie z danych niezbędnych informacji



Informacja

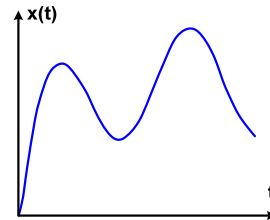
- Co oznaczają poniższe dane?



Informacja analogowa i cyfrowa

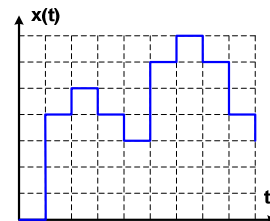
■ Sygnał analogowy

- może przyjmować dowolną wartość z ciągłego przedziału (nieskończonego lub ograniczonego zakresem zmienności)
- wartości mogą zostać określone w każdej chwili czasu dzięki funkcji matematycznej opisującej dany sygnał



■ Sygnał cyfrowy

- dziedziina i zbiór wartości są dyskretne
- sygnał ciągły, który może zmieniać swoją wartość tylko w określonych chwilach czasu i może przyjmować tylko określone wartości



Liczby i cyfry

■ Liczba - pojęcie abstrakcyjne, abstrakcyjny wynik obliczeń, wartość

- umożliwia wyrażenie wyniku liczenia przedmiotów oraz mierzenia wielkości

■ Cyfra - umowny znak (symbol) stosowany do zapisu liczby

- liczba znaków służących do zapisu jest zależna od systemu liczbowego i przyjętego sposobu zapisu
- system dziesiętny - 10 znaków
- system szesnastkowy - 16 znaków
- system rzymski - 7 znaków

■ Cyfry rzymskie

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Informacja analogowa i cyfrowa

■ Zalety sygnałów cyfrowych:

- odporne na zakłócenia
- powtarzalne (np. kopia filmu na DVD i VHS)
- możliwość przesyłania na duże odległości
- możliwość szyfrowania sygnału (kryptografia)
- niższe koszty przetwarzania

■ Wady sygnałów cyfrowych:

- ograniczenie częstotliwości próbkowania (sygnał analogowy zamieniony na cyfrowy i ponownie na analogowy nie jest już tym samym sygnałem)

Liczby i cyfry

■ Cyfry arabskie (pochodzą z Indii)

- arabskie, standardowe europejskie

1	2	3	4	5	6	7	8	9	0
---	---	---	---	---	---	---	---	---	---

- indyjsko-arabskie

١	٢	٣	٤	٥	٦	٧	٨	٩	٠
1	2	3	4	5	6	7	8	9	0

- wschodnio-indyjsko-arabskie

١	٢	٣	٤	٥	٦	٧	٨	٩	٠
1	2	3	4	5	6	7	8	9	0

■ W niektórych systemach jako cyfry stosowane są litery, np.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:

cyfry etruskie
 $I \cdot \wedge \cdot X \cdot XX \cdot \wedge XX \cdot \uparrow \cdot * (C) \cdot \odot (\Phi)$
 1 5 10 20 25 50 100 1000

cyfry grecko-jońskie

1	2	3	4	5	6	7	8	9
α	β	γ	δ	ε	ς	ζ	η	θ
10	20	30	40	50	60	70	80	90
ι	κ	λ	μ	ν	ξ	ο	π	ρ
100	200	300	400	500	600	700	800	900
σ	τ	υ	φ	χ	ψ	ω	Ϡ	ϡ
1000	2000	3000	4000	5000	6000	7000	8000	9000
Ϡ	β	γ	δ	ε	ς	ζ	η	θ
10000	20000	30000	40000	50000	60000	70000	80000	90000
Ϡ	ϡ	β	γ	δ	ε	ς	ζ	η

cyfry w pisowni chińskiej

jeden	一	sześć	六
dwa	二	siedem	七
trzy	三	osiem	八
cztery	四	dziewięć	九
pieć	五	dziesięć	十
zero	另		

uczniestehinskiego.com

Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:

liczby w piśmie klinowym (Babilończycy)

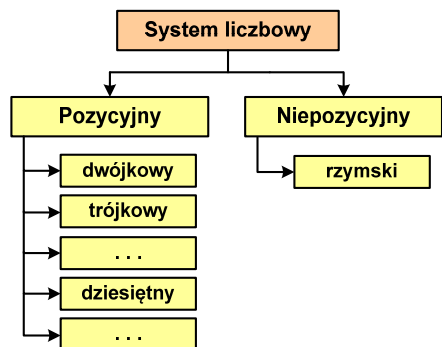
1 2 3 4
 5 6 7
 8 9
 10 20 30
 100 1000

system prekolumbijski

0 1 2 3 4
 5 6 7 8 9
 10 11 12 13 14
 15 16 17 18 19

Systemy liczbowe

- System liczbowy** - zbiór zasad umożliwiających zapis liczb za pomocą cyfr oraz wykonywanie działań na tych liczbach



- Pozycyjny** - znaczenie cyfry jest zależne od miejsca (pozycji), które zajmuje ona w liczbie
 - system dziesiętny - liczba 111 (każda cyfra ma inne znaczenie)
- Niepozycyjny** - znaczenie cyfry jest niezależne od miejsca położenia w liczbie
 - system rzymski - liczba III

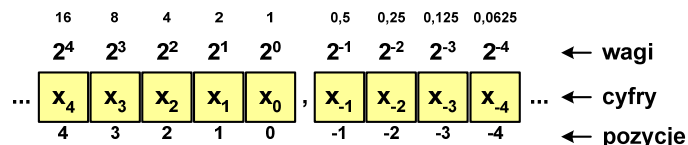
System dziesiętny (ang. decimal)

10000	1000	100	10	1	0,1	0,01	0,001	0,0001		
10^4	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	← wagi	
...	x_4	x_3	x_2	x_1	x_0	x_{-1}	x_{-2}	x_{-3}	x_{-4}	...
	4	3	2	1	0	-1	-2	-3	-4	← pozycje

- p - podstawa systemu pozycyjnego, D - zbiór dozwolonych cyfr
- $p = 10$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$$\begin{aligned}
 & \begin{matrix} 10^3 & 10^2 & 10^1 & 10^0 & 10^{-1} & 10^{-2} \\ \boxed{1} & \boxed{4} & \boxed{0} & \boxed{8} & \boxed{2} & \boxed{5} \end{matrix} \\
 1408,25_{(10)} = & \\
 = & \boxed{1 \cdot 10^3} + \boxed{4 \cdot 10^2} + \boxed{0 \cdot 10^1} + \boxed{8 \cdot 10^0} + \boxed{2 \cdot 10^{-1}} + \boxed{5 \cdot 10^{-2}} \\
 = & 1000 + 400 + 0 + 8 + 0,2 + 0,05
 \end{aligned}$$

System dwójkowy (ang. binary)

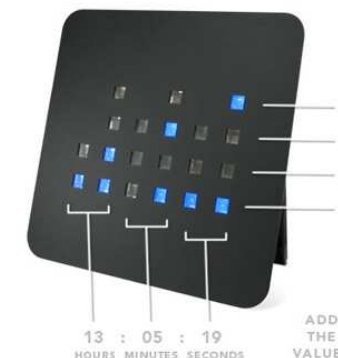


- w systemie dwójkowym: $p = 2$, $D = \{0, 1\}$

$$\begin{aligned}
 & \begin{array}{cccccccc} 2^3 & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} & \\ \hline & 1 & 1 & 0 & 1 & , & 1 & 0 & 1 \end{array} \\
 1101,101_{(2)} &= \\
 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\
 &= 8 + 4 + 0 + 1 + 0,5 + 0 + 0,125 \\
 &= 13,625_{(10)}
 \end{aligned}$$

System dwójkowy - zastosowania

- Powszechnie używany w informatyce, technice cyfrowej



System szesnastkowy (ang. hexadecimal)

- System heksadecymalny
- $p = 16$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Powszechnie używany w informatyce - jeden bajt można zapisać za pomocą tylko dwóch cyfr szesnastkowych

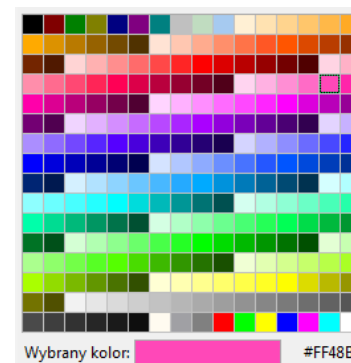
$$3A5D_{(16)} = 3 \cdot 16^3 + 10 \cdot 16^2 + 5 \cdot 16^1 + 13 \cdot 16^0 = 14941_{(10)}$$

- Sposoby zapisu liczb w systemie szesnastkowym:

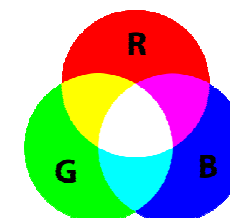
3A5Dh	0x3A5D	#3A5D
3A5D ₍₁₆₎	3A5D ₁₆	3A5D _{hex}
(3A5D) _{hex}	(3A5D) ₁₆	\$3A5D

System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Każda barwa przyjmuje wartość z zakresu: $0..255_{(10)}$, $00..FF_{(16)}$



#FF48B8



System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Kolory w dokumentach HTML:

```
<BODY bgcolor="#336699" text="#000000" link="#FFFF00"  
vlink="#33FFFF" alink="#FF0000">
```

ARCHIWUM

ENGLISH

Studia stacjonarne:

Poniedziałek:
12:15 - 14:00 [Informatyka 1 - wykład](#), sem. 2 ED, WE-Aula II

Wtorek:
08:30 - 10:00 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS3, WE-110
12:15 - 13:45 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS2, WE-110
14:15 - 15:45 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS1, WE-110

Przykład systemu niepozycyjnego - system rzymski

- W systemie rzymskim posługujemy się siedmioma znakami:
I - 1 V - 5 X - 10 L - 50 C - 100 D - 500 M - 1000
- Za pomocą dostępnych symboli można określić liczby od 1 do 3999
- System **addytywny** - wartość liczby określa się na podstawie sumy wartości cyfr, np.
 - II (1 + 1 = 2), XXX (10 + 10 + 10 = 30)
 - CLX (100 + 50 + 10 = 160), MMXII (1000 + 1000 + 10 + 1 + 1 = 2012)
- Wyjątkiem od powyższej zasady są liczby do opisu których używa się odejmowania, np.
 - IV (5 - 1 = 4), IX (10 - 1 = 9), XL (50 - 10 = 40), XC (100 - 10 = 90)
- Stosowany w łacińskiej części Europy do końca Średniowiecza
- Niewygodny w prowadzeniu nawet prostych działań arytmetycznych, brak ułamków

System szesnastkowy - zastosowania

- 48-bitowy adres fizyczny urządzenia (MAC - Media Access Control)

88:AD:D2:09:41:3B

producent numer egzemplarza

- <http://hwaddress.com>

HWAddress Home Countries Companies 88:AD:D2 Search

OUI	MAC range	Company
88-AD-D2	88-AD-D2-00-00-00 - 88-AD-D2-FF-FF-FF	Samsung Electronics Co.,Ltd

Przykład systemu niepozycyjnego - system rzymski

- Zasady tworzenia liczb:
 - zestawiamy odpowiednie znaki od oznaczającego liczbę największą do oznaczającego liczbę najmniejszą
 $XVI = 10(X) + 5(V) + 1(I) = 16$
 - jeżeli składnik liczby, którą piszemy, jest wielokrotnością liczby nominalnej, wtedy zapisywany jest z użyciem kilku następujących po sobie znaków
 $CCC = 100(C) + 100(C) + 100(C) = 300$
 - dodatkowo należy zachować zasadę nie pisania czterech tych samych znaków po sobie, lecz napisać jeden znak wraz ze znakiem oznaczającym wartość większą o jeden rząd liczbowy
 $CD = 500(D) - 100(C) = 400$

Przykład systemu niepozycyjnego - system rzymski

■ Zasady odczytu liczb:

- cyfry jednakowe są dodawane

$$MMM = 1000(M) + 1000(M) + 1000(M) = 3000$$

- cyfry mniejsze stojące przed większymi są odejmowane od nich

$$CDXCIV = 500(D) - 100(C) + 100(C) - 10(X) + 5(V) - 1(I) = 494$$

- cyfry mniejsze stojące za większymi są do nich dodawane

$$MDCLX = 1000(M) + 500(D) + 100(C) + 50(L) + 10(X) = 1660$$

Konwersja na system dziesiętny

- $p = 4, D = \{0, 1, 2, 3\}$

$$21302_{(4)} = ?_{(10)}$$

$$\begin{matrix} 4^4 & 4^3 & 4^2 & 4^1 & 4^0 \\ 2 & 1 & 3 & 0 & 2 \end{matrix}$$

$$21302_{(4)} = 2 \cdot 4^0 + 0 \cdot 4^1 + 3 \cdot 4^2 + 1 \cdot 4^3 + 2 \cdot 4^4$$

$$21302_{(4)} = 2 \cdot 1 + 0 \cdot 4 + 3 \cdot 16 + 1 \cdot 64 + 2 \cdot 256$$

$$21302_{(4)} = 2 + 0 + 48 + 64 + 512 = 626_{(10)}$$

- $p = 17, D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G\}$

$$AC24_{(17)} = ?_{(10)}$$

$$\begin{matrix} 17^3 & 17^2 & 17^1 & 17^0 \\ A & C & 2 & 4 \end{matrix}$$

$$AC24_{(17)} = 4 \cdot 17^0 + 2 \cdot 17^1 + 12 \cdot 17^2 + 10 \cdot 17^3$$

$$AC24_{(17)} = 4 \cdot 1 + 2 \cdot 17 + 12 \cdot 289 + 10 \cdot 4913$$

$$AC24_{(17)} = 4 + 34 + 3468 + 49130 = 52636_{(10)}$$

Konwersja na system dziesiętny (schemat Hornera)

- $p = 4, D = \{0, 1, 2, 3\}$

$$21302_{(4)} = w_{(10)} \quad x_4 x_3 x_2 x_1 x_0 = w_{(10)}$$

$$w_{(10)} = 0$$

$$w_{(10)} = x_4 + w_{(10)} \cdot p = 2 + 0 \cdot 4 = 2$$

$$w_{(10)} = x_3 + w_{(10)} \cdot p = 1 + 2 \cdot 4 = 9$$

$$w_{(10)} = x_2 + w_{(10)} \cdot p = 3 + 9 \cdot 4 = 39$$

$$w_{(10)} = x_1 + w_{(10)} \cdot p = 0 + 39 \cdot 4 = 156$$

$$w_{(10)} = x_0 + w_{(10)} \cdot p = 2 + 156 \cdot 4 = 626_{(10)}$$

Konwersja z systemu dziesiętnego na dowolny

- zamiana liczby z systemu $p = 10$ na system $p = 2$

$$626_{(10)} = ?_{(2)}$$

$$626_{(10)} = 1001110010_{(2)}$$

$$626/2 = 313 \text{ reszta } 0$$

$$313/2 = 156 \text{ reszta } 1$$

$$156/2 = 78 \text{ reszta } 0$$

$$78/2 = 39 \text{ reszta } 0$$

$$39/2 = 19 \text{ reszta } 1$$

$$19/2 = 9 \text{ reszta } 1$$

$$9/2 = 4 \text{ reszta } 1$$

$$4/2 = 2 \text{ reszta } 0$$

$$2/2 = 1 \text{ reszta } 0$$

$$1/2 = 0 \text{ reszta } 1$$

kolejność odczytywania
cyfr liczby w systemie
dwójkowym

kończymy, gdy liczba dziesiętna ma wartość 0

Konwersja z systemu dziesiętnego na dowolny

- zamiana liczby z systemu $p = 10$ na system $p = 7$

$$626_{(10)} = ?_{(7)} \qquad 626_{(10)} = 1553_{(7)}$$

$626/7 = 89$	reszta 3	↑
$89/7 = 12$	reszta 5	
$12/7 = 1$	reszta 5	
$1/7 = 0$	reszta 1	

- zamiana liczby z systemu $p = 10$ na system $p = 14$

$$626_{(10)} = ?_{(14)} \qquad 626_{(10)} = 32A_{(14)}$$

$626/14 = 44$	reszta 10	↑ → A
$44/14 = 3$	reszta 2	
$3/14 = 0$	reszta 3	

Szybkie konwersje: $2 \rightarrow 4, 8, 16$ $4, 8, 16 \rightarrow 2$

$2 \rightarrow 4$

$$110110011_{(2)} = ?_{(4)}$$

01	10	11	00	11
1	2	3	0	3

$$110110011_{(2)} = 12303_{(4)}$$

$2 \rightarrow 8$

$$10110011_{(2)} = ?_{(8)}$$

010	110	011
2	6	3

$$10110011_{(2)} = 263_{(8)}$$

$2 \rightarrow 16$

$$1011010_{(2)} = ?_{(16)}$$

0101	1010
5	A

$$1011010_{(2)} = 5A_{(16)}$$

$4 \rightarrow 2$

$$12303_{(4)} = ?_{(2)}$$

1	2	3	0	3
01	10	11	00	11

$$12303_{(4)} = 110110011_{(2)}$$

$8 \rightarrow 2$

$$263_{(8)} = ?_{(2)}$$

2	6	3
010	110	011

$$263_{(8)} = 10110011_{(2)}$$

$16 \rightarrow 2$

$$5A_{(16)} = ?_{(2)}$$

5	A
0101	1010

$$5A_{(16)} = 1011010_{(2)}$$

Koniec wykładu nr 1

Dziękuję za uwagę!