



Politechnika Białostocka
Wydział Elektryczny
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki

Instrukcja
do pracowni specjalistycznej z przedmiotu

**Programowanie mikrokontrolerów
w języku wysokiego poziomu 1**

Kod przedmiotu: **TS1F1008**

(studia stacjonarne)

ARDUINO - TABLICE W JĘZYKU C

Numer ćwiczenia

PMC_11

Autor:
dr inż. Jarosław Forenc

Białystok 2024

Spis treści

1. Opis stanowiska	3
1.1. Stosowana aparatura	3
1.2. Oprogramowanie.....	3
2. Wiadomości teoretyczne.....	3
2.1. Tablica elementów	3
2.2. Tablica jednowymiarowa (wektor)	4
2.3. Generowanie pseudolosowe elementów tablicy	6
2.4. Inicjalizacja elementów tablicy.....	8
3. Przebieg ćwiczenia.....	9
4. Literatura.....	10
5. Pytania kontrolne	10
6. Wymagania BHP	11

Materiały dydaktyczne przeznaczone dla studentów Wydziału Elektrycznego PB.

© Wydział Elektryczny, Politechnika Białostocka, 2024 (wersja 1.1)

Wszelkie prawa zastrzeżone. Żadna część tej publikacji nie może być kopiowana i odtwarzana w jakiegokolwiek formie i przy użyciu jakichkolwiek środków bez zgody posiadacza praw autorskich.

1. Opis stanowiska

1.1. Stosowana aparatura

Podczas zajęć wykorzystywany jest komputer klasy PC z systemem operacyjnym Microsoft Windows 10/11 oraz platforma Arduino wraz z zestawem czujników.

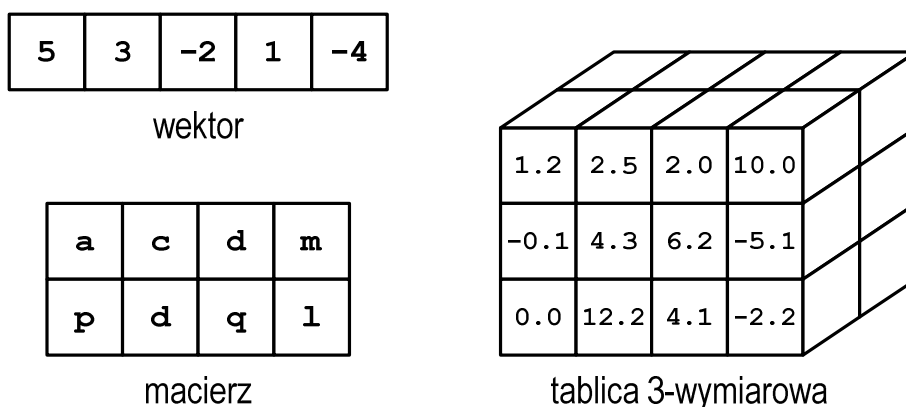
1.2. Oprogramowanie

Na komputerach zainstalowany jest edytor kodu źródłowego Visual Studio Code 1.92 (lub nowszy) wraz z rozszerzeniem (PlatformIO IDE for VSCode).

2. Wiadomości teoretyczne

2.1. Tablica elementów

Tablica elementów jest ciągłym obszarem pamięci, w którym te elementy są przechowywane. W tablicy mogą znajdować się elementy tylko jednego typu. Wyróżnia się tablice jednowymiarowe (wektory), dwuwymiarowe (macierze) oraz tablice o większej liczbie wymiarów (Rys. 1).



Rys. 1. Tablice elementów w języku C

Głównym celem stosowania tablic jest zastąpienie wielu zmiennych tego samego typu jedną tablicą.

2.2. Tablica jednowymiarowa (wektor)

Deklarując tablicę jednowymiarową, należy podać: typ elementów, nazwę tablicy oraz liczbę jej elementów, np.

```
int tab[5];
```

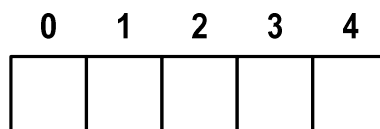
↑
↑
↑
↑
↑

średnik
rozmiar tablicy
nazwa tablicy
typ elementów tablicy

Wyrażenie podane w nawiasach kwadratowych, określające rozmiar tablicy, musi dawać w wyniku dodatnią stałą całkowitoliczbową. Ponadto wartość ta musi być znana już w fazie kompilacji (nie może to być zmienna). Jako rozmiar tablicy można także podać nazwę stałej zdefiniowanej dyrektywą preprocesora **#define** lub za pomocą słowa kluczowego **const**.

<pre>int tab[5];</pre>	<pre>#define N 5 int tab[N];</pre>	<pre>const int n = 5; int tab[n];</pre>
------------------------	--	---

Powyższe deklaracje definiują tablicę pięciu elementów typu `int` (Rys. 2). Jest to tablica jednowymiarowa, czyli tzw. **wektor**.

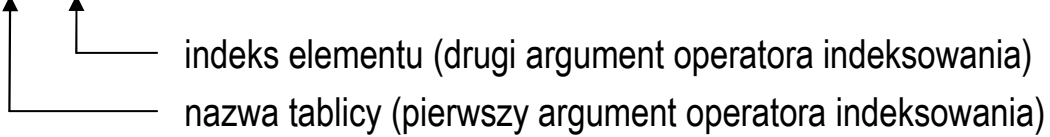


Rys. 2. Wektor 5-elementowy

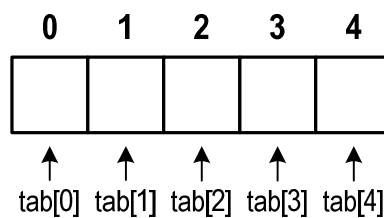
Każdy element tablicy ma swój numer zwany **indeksem**. Element znajdujący się na początku tablicy ma indeks **0** (zero), natomiast ostatni element ma indeks **N-1**, gdzie **N** to rozmiar tablicy. Wartość indeksu wskazuje, o ile elementów dany element jest odległy od początku tablicy. Nazwa tablicy reprezentuje adres jej zerowego elementu (o indeksie **0**) w pamięci komputera.

Odwołania do elementów tablicy (odczytanie lub zapisanie wartości) wykonuje się za pomocą dwuargumentowego operatora indeksowania [], np.

`tab[1]` - odwołanie do elementu tablicy o indeksie 1



Odwołania do kolejnych elementów tablicy `tab` mają postać pokazaną na Rys. 3.



Rys. 3. Odwołania do elementów tablicy

Zapisanie wartości `5` do elementu tablicy `tab` o indeksie `1` oraz odczytanie tego elementu i przypisanie jego wartości zmiennej o nazwie `x`:

```
tab[1] = 5;  
x = tab[1];
```

Jako indeks może występować:

- stała liczbowa, np. `0`, `1`, `5`;
- nazwa zmiennej przechowującej liczbę całkowitą, np. `i`, `idx`;
- wyrażenie dające w wyniku liczbę całkowitą, np. `i*j+5`.

Przy odwołaniach do elementów tablicy kompilator nie sprawdza, czy zapis lub odczyt odbywa się w obszarze pamięci przydzielonym na tablicę, np.

```
int tab[5];  
tab[5] = 10;
```

W powyższym fragmencie programu zadeklarowano 5-elementową tablicę o nazwie `tab`. Odwołanie `tab[5]` jest błędne, ponieważ indeks `5` jest poza zakresem

tablicy. Kompilator nie zasygnalizuje błędu, a w obszarze pamięci za tablicą zapisze wartość **10**.

Operacje na tablicach najczęściej wykonuje się przy wykorzystaniu pętli **for**. Załóżmy, że do wszystkich elementów tablicy **tab** należy zapisać wartość **10**. Kod realizujący taką operację może mieć następującą postać:

```
int tab[5];

tab[0] = 10;
tab[1] = 10;
tab[2] = 10;
tab[3] = 10;
tab[4] = 10;
```

Można to samo zrobić znacznie prościej, stosując pętlę **for**:

```
int tab[5], i;

for (i=0; i<5; i++)
    tab[i] = 10;
```

Zmienna **i** przyjmuje wartości od **0** do **4**, czyli takie same, jak kolejne indeksy elementów tablicy.

2.3. Generowanie pseudolosowe elementów tablicy

Elementy tablicy mogą być wygenerowane pseudolosowo, co ilustruje poniższy program.

Generowanie pseudolosowe elementów tablicy.

```
#include <Arduino.h>

#define N 10
```

```

void setup() {
  Serial.begin(9600);
  srand(analogRead(0));
}

void loop() {
  int  tab[N], i;

  for (i=0; i<N; i++) {
    tab[i] = rand();
  }

  /* wyświetlenie elementów tablicy */

  Serial.println("Elementy tablicy:");
  for (i=0; i<N; i++) {
    Serial.print(tab[i]);
    Serial.print(" ");
  }
  Serial.println(" ");

  delay(5000);
}

```

Do generowania pseudolosowych liczb zastosowano funkcję **rand()**:

```
tab[i] = rand();
```

Funkcja ta zwraca pseudolosową liczbę całkowitą z zakresu od 0 do **RAND_MAX** (zwykle 32767). Przed użyciem funkcji **rand()** należy zainicjalizować generator liczb pseudolosowych, wywołując funkcję **srand()**:

```
srand(analogRead(0));
```

której argumentem jest liczba inicjalizująca generator. Aby zapewnić unikalność generowania kolejnych liczb, do funkcji **srand()** przekazywana jest wartość odczytana z niepodłączonego pinu wejściowego modułu Arduino.

Zmiana zakresu generowanych liczb odbywa się poprzez zastosowanie dzielenia modulo. Jeśli chcemy otrzymać liczby całkowite z zakresu od **0** do **10**, to wystarczy wartość zwracaną przez funkcję **rand()** podzielić **modulo 11**:

```
int x;  
x = rand() % 11;
```

Pseudolosową liczbę całkowitą z przedziału **[a, b]** otrzymamy, używając funkcji **rand()** w następujący sposób:

```
int x;  
x = rand() % (b - a + 1) + a;
```

2.4. Inicjalizacja elementów tablicy

Po zadeklarowaniu tablicy wartości jej elementów są nieokreślone. Inicjalizacja elementów tablicy polega na nadaniu wartości poszczególnym elementom od razu przy deklaracji. Taka inicjalizacja polega na umieszczeniu po znaku równości, ujętej w nawiasy klamrowe, listy wartości dla kolejnych elementów, np.

```
int a[3] = {5,7,1};
```

0	1	2
5	7	1

Poszczególne elementy tablicy są oddzielone od siebie przecinkami. Jako kolejne elementy mogą występować liczby lub wyrażenia arytmetyczne. Tablice można inicjalizować **tylko** podczas deklaracji.

Jeśli wartości podanych w trakcie inicjalizacji jest mniej niż rozmiar tablicy, to pozostałe elementy tablicy są wypełniane zerami, np.

```
int a[3] = {5,7,1};
```

0	1	2	3	4
5	7	1	0	0

Jeśli liczba wartości podanych w trakcie inicjalizacji jest więcej niż wynosi rozmiar tablicy, to kompilator zasygnalizuje błąd, np.

```
int a[5] = {5,7,1,3,6,4};
```

Tablica zadeklarowana bez podania rozmiaru, ale zainicjalizowana, ma liczbę elementów równą liczbie wartości inicjalizujących, np.

```
int a[] = {2,3,1,4};
```

jest równoważne:

```
int a[4] = {2,3,1,4};
```

3. Przebieg ćwiczenia

Na pracowni specjalistycznej należy wykonać wybrane zadania wskazane przez prowadzącego zajęcia. W różnych grupach mogą być wykonywane różne zadania.

1. Zadeklaruj **20**-elementową tablicę liczb całkowitych. Wypełnij tablicę liczbami pseudolosowymi z zakresu od **1** do **4**. Każda liczba w tablicy odpowiada numerowi diody LED (**1** - **LED1**, **2** - **LED2**, **3** - **LED3**, **4** - **LED4**). Odczytuj kolejne liczby z tablicy i zapalaj odpowiadające im diody LED. Czas świecenia każdej diody LED powinien wynosić 500 ms.
2. Zadeklaruj 10-elementową tablicę do przechowywania wyników pomiaru temperatury czujnikiem **BMP280**. Zapisanie kolejnej wartości do tablicy powinno nastąpić po naciśnięciu przycisku **SW1**. Po zapisaniu **dziesiątej** wartości, zakończenie pomiarów powinno zostać zasygnalizowane dźwiękiem buzzera, a na monitorze portu szeregowego powinna zostać wyświetlona najniższa i najwyższa zmierzona temperatura. Naciśnięcie, w dowolnym

momencie, przycisku **SW2** powinno spowodować usunięcie wszystkich dotychczas zapamiętanych wartości.

3. Napisz program, którego zadaniem jest wyznaczenie średniego natężenia światła na biurku, przy którym pracujesz. Podczas pracy programu naciśnięcie przycisku **SW1** powinno spowodować zapamiętanie bieżącego natężenia oświetlenia, natomiast naciśnięcie przycisku **SW2** powinno uruchomić obliczenie średniego natężenia oświetlenia i wyświetlenie wyniku na wyświetlaczu **OLED**. Wyniki pomiarów należy zapisać w tablicy, przy czym maksymalna liczba pomiarów wynosi **20**. Stosując napisany program, zmierz natężenie oświetlenia w różnych punktach biurka, a następnie wyświetl średnie natężenie.

4. Literatura

- [1] Prata S.: Język C. Szkoła programowania. Wydanie VI. Helion, Gliwice, 2016.
- [2] Wrotek W.: Arduino od podstaw. Helion, Gliwice, 2023.
- [3] Monk S.: Arduino dla początkujących. Podstawy i szkice. Helion, Gliwice, 2019.
- [4] Evans M., Noble J., Hochenbaum J.: Arduino w akcji. Helion, Gliwice, 2014.
- [5] Kernighan B.W., Ritchie D.M.: Język ANSI C. Programowanie. Wydanie II. Helion, Gliwice, 2010.
- [6] <https://code.visualstudio.com/> - Visual Studio Code
- [7] <https://www.arduino.cc/reference/en/> - Arduino Language Reference

5. Pytania kontrolne

1. Omów sposób deklaruowania tablic jednowymiarowych (wektorów) w języku C oraz sposób odwoływania się do elementów tych tablic.
2. Wyjaśnij, jak odszukać w tablicy jednowymiarowej element o najmniejszej i element o największej wartości.

3. W jaki sposób w języku C można generować pseudolosowe liczby całkowite i rzeczywiste z określonego zakresu?
4. Opisz sposób inicjalizacji elementów tablicy jednowymiarowej.

6. Wymagania BHP

Warunkiem przystąpienia do praktycznej realizacji ćwiczenia jest zapoznanie się z instrukcją BHP i instrukcją przeciwpożarową oraz przestrzeganie zasad w nich zawartych.

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad.

- Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.
- Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor komputera ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
- Sprawdzić prawidłowość połączeń urządzeń.
- Załączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
- W trakcie pracy z komputerem zabronione jest spożywanie posiłków i picie napojów.
- W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
- Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
- Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.

- W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
- Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.
- Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
- W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie stanowiska. Przed odłączeniem napięcia nie dotykać porażonego.