

MoboLab – roboty i tablety w Twojej szkole

Obszar II. „Stwórz własnego robota”

Scenariusze lekcji i zajęć pozalekcyjnych

SCENARIUSZ 22. PROGRAMOWANIE KLAWIATURY MEMBRANOWEJ

scenariusz zajęć pozalekcyjnych

autor: Michał Podziomek

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

Arduino, programowanie, elektronika, klawiatura, przyciski, włączniki

KRÓTKI OPIS ZAJĘĆ:

Podczas zajęć uczniowie i uczennice uczą się, jak stworzyć układ elektroniczny za pomocą Arduino oraz **klawiatury membranowej**. Przygotowują układ i tworzą program pozwalający na odczyt wartości z klawiatury membranowej.

WIEDZA I UMIEJĘTNOŚCI ZDOBYTE PRZEZ UCZNIĄ / UCZENNICĘ:

- wie, czym są mikrokontrolery i do czego służą,
- zna pojęcia: mikrokontroler, skrypt, program, algorytm, sterowanie, warunek, pętla,
- zna projekt Arduino, wie, czym jest platforma Arduino, z jakich części się składa,
- potrafi w podstawowym stopniu samodzielnie obsługiwać Arduino (podłączyć płytkę do komputera, wgrać prosty program),
- zna podstawowe elementy interfejsu środowiska programistycznego Arduino IDE i podstawowe komendy języka Arduino IDE: `pinMode()`, `digitalWrite()`, `delay()`,
- rozumie zasadę działania funkcji `digitalWrite()` i potrafi wykorzystać ją w praktyce,
- wie, że istnieją sposoby podłączenia wielu sygnałów do pojedynczego pinu, rozumie, na czym polega multipleksowanie,
- zna podstawowe elementy języka **Scratch**, potrafi stworzyć prosty skrypt w tym języku.

GRUPA DOCELOWA:

Starsze klasy szkoły podstawowej (VII-) i klasy gimnazjalne (po dostosowaniu: możliwość realizacji w młodszych klasach: I-III i IV-VI szkoły podstawowej). W młodszych klasach – możliwość wykorzystania programu mBlock (po przejściu scenariusza nr 18. *Programowanie Arduino z wykorzystaniem programu mBlock*) lub Scratch for Arduino (po przejściu scenariusza nr 1. *Wprowadzenie do Arduino*).

LICZBA UCZNIÓW/UCZENNIC W GRUPIE:

Liczba optymalna: 12, liczba maksymalna: 16

CZAS TRWANIA ZAJĘĆ:

90 min (lub 2 x 45 minut)

STOPIEŃ TRUDNOŚCI/SKOMPLIKOWANIA

(w skali od 1 do 5 dla obszaru II. „Stwórz własnego robota”):

2

POTRZEBNY SPRZĘT I OPROGRAMOWANIE:

- komputer (przenośny lub stacjonarny),
- program Arduino IDE (do pobrania ze strony: <http://www.arduino.org/downloads>),
- (opcjonalnie) program mBlock (do pobrania ze strony: <http://www.mblock.cc/download/>) lub Scratch for Arduino (do pobrania ze strony: <http://s4a.cat/>),
- płytki Arduino UNO i kabel USB A-B (dla każdego uczestnika lub dla pary uczestników),
- płytki stykowe, przewody połączeniowe,
- klawiatura membranowa 4x4,
- brzęczyk
- projektor i laptop (w części teoretycznej).

CO NALEŻY PRZYGOTOWAĆ PRZED ZAJĘCIAMI:

- zainstalować program Arduino IDE,
- (opcjonalnie): zainstalować program **mBlock** lub **Scratch for Arduino**,
- sprawdzić, czy wszystkie komputery wykrywają podłączone Arduino,
- przeczytać dokładnie scenariusz, zapoznać się z materiałami dodatkowymi (w części „Pigułka wiedzy i inspiracji”),
- wykonać samodzielnie zadania zawarte w scenariuszu,

- przy każdym stanowisku komputerowym rozłożyć elementy zestawu Arduino, które będą wykorzystywane na tych zajęciach,
- dopasować stopień trudności zadania do potrzeb i możliwości klasy, dla której organizowane są zajęcia według wskazówek zawartych w scenariuszu.

KOMPETENCJE OSOBY PROWADZĄCEJ:

- wie, czym jest projekt Arduino, zna podstawowe informacje o projekcie,
- potrafi przynajmniej w stopniu podstawowym obsługiwać Arduino,
- zna podstawowe pojęcia z zakresu elektroniki,
- zna podstawowe pojęcia programistyczne,
- wie, dlaczego warto uczyć się programowania i jakie korzyści daje posiadanie umiejętności programistycznych,
- potrafi zachęcić do nauki programowania zarówno chłopców, jak i dziewczynki.

PRZEBIEG ZAJĘĆ:

Podłączenie Arduino, uruchomienie programu Arduino IDE i przypomnienie podstawowych informacji – ok. 15 minut

Uwaga! Informacje o tym, jak podłączyć Arduino, uruchomić program Arduino IDE i Scratch for Arduino, a także podstawowe informacje niezbędne przy rozpoczynaniu pracy z Arduino zawierają scenariusze 1 i 2. Tę część zajęć warto powtarzać za każdym razem w takim zakresie, jaki jest potrzebny, do czasu aż podstawowy materiał zostanie utrwalony.

Omawiamy sposób podłączania klawiatury do Arduino – 15 minut

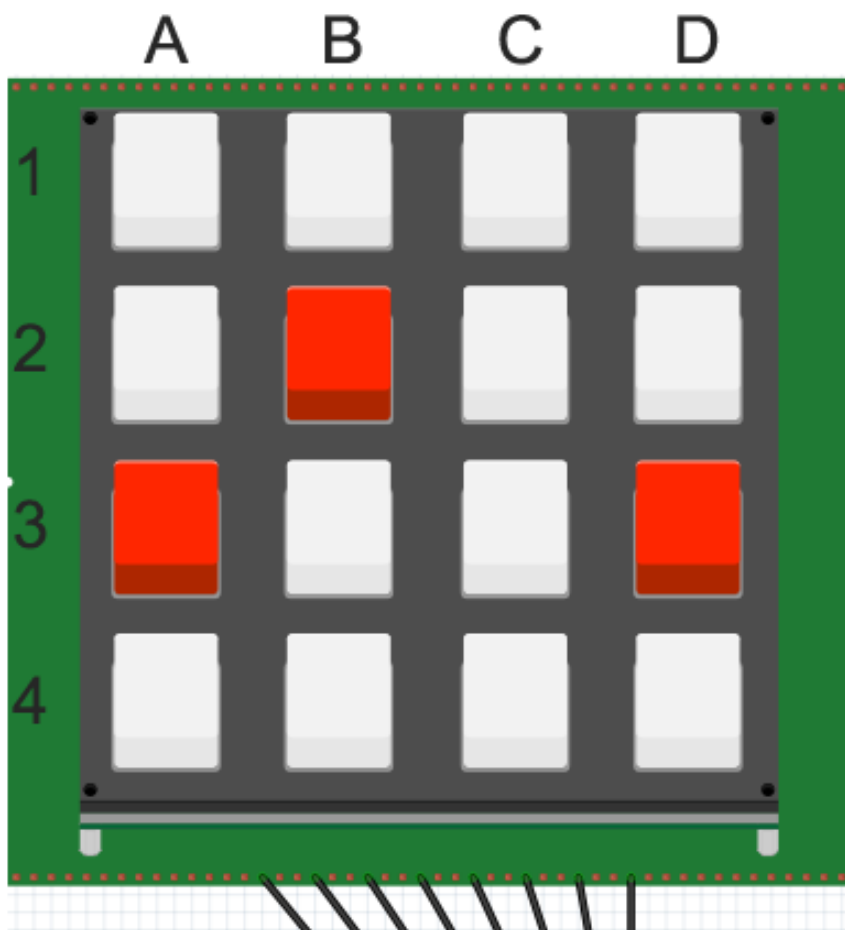
Zajęcia są poświęcone klawiaturze membranowej i wykorzystywaniu jej w układach elektronicznych z Arduino. Rozpoczynamy od omówienia klawiatury oraz sposobu podłączania jej do Arduino.

Prosimy uczniów o policzenie przycisków na klawiaturze, a następnie o policzenie pinów, do których można ją podłączyć.

Oczekiwana odpowiedź: **pinów jest mniej niż przycisków**. Jakie można z tego wyciągnąć wnioski?

Do tej pory podłączaliśmy po jednym przycisku do pinu i odczytywaliśmy sygnał z niego płynący. Dla urządzeń o dużej liczbie przycisków, jak np. klawiatury komputerowe, byłoby to niepraktyczne, ponieważ musielibyśmy stosować wiele mikrokontrolerów, żeby obsłużyć tak dużo przycisków, lub stosować bardzo drogie wielopinowe kontrolery. Dodatkowym problemem byłaby liczba przewodów biegnących między mikrokontrolerem a każdym przełącznikiem. Zostały więc wymyślone sposoby na to, jak do mniejszej liczby pinów podłączyć wiele urządzeń.

Najprostszym z nich jest naprzemienne odczytywanie danych ze ścieżek ułożonych w kolumny i wiersze. Adresowanie to jest bardzo podobne do tego używanego w notacji szachowej: kolumny są oznaczone literami, a wiersze cyframi. Dla naszej klawiatury wyglądałoby to w sposób następujący:



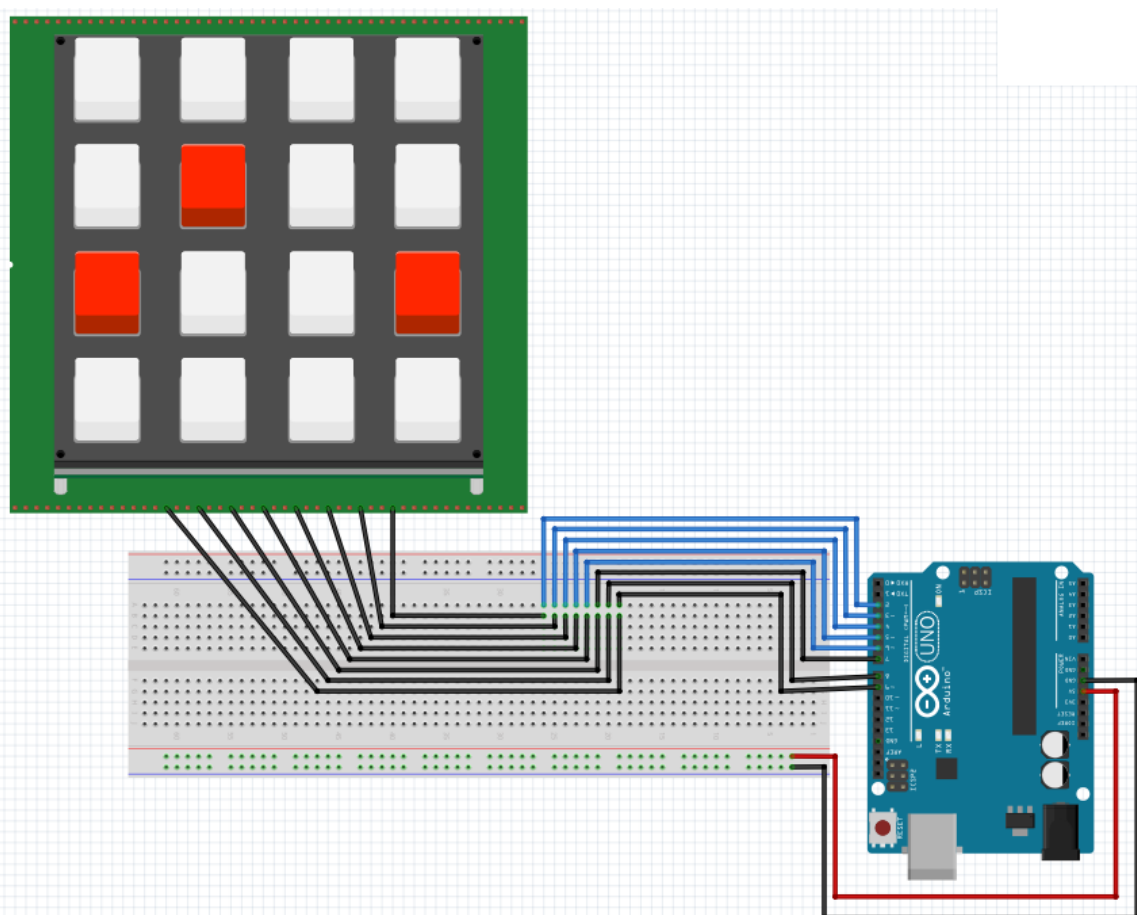
Do płytki jest podłączone sumarycznie 8 przewodów, które obsługuje 16 przycisków. 4 z nich są poprowadzone przez wiersze 1-4, a kolejne 4 przez kolumny A-D. Włączając przełącznik, zwieramy ze sobą jeden przewód kolumny i jeden

przewód ścieżki i stosując odpowiedni program, jesteśmy w stanie odczytać który z przycisków jest wciśnięty.

W praktyce wygląda to tak, że mikroprocesor z bardzo dużą częstotliwością wysyła prąd naprzemiennie do każdej z kolumn (pojedynczo) i sprawdza, na którym pinie wiersza ten sygnał do niego wróci. Tak więc, jeśli wyślemy sygnał do kolumny C i wróci on do naszej elektroniki na pinie 3, to oznacza że mamy wciśnięty przycisk C3. Po podłączeniu naszego układu elektronicznego spróbujemy to przedstawić w formie kodu.

Montujemy układ – 15 minut

Montujemy układ zgodnie z poniższym schematem:



Patrząc się na klawiaturę z góry, jej pierwszy lewy pin będzie odpowiadał pinowi 2 na Arduino, a ostatni prawy – pinowi 9. Pierwsze 4 piny to wiersze, kolejne 4 to

kolumny. Jeżeli jest taka możliwość, możemy podpiąć klawiaturę do Arduino bez wykorzystania płytki stykowej.

W tym miejscu możliwy jest podział zajęć na dwie części (kolejna część scenariusza będzie realizowana na następnych zajęciach).

Przypomnienie materiału, odtworzenie układu – 10 minut

Rozpoczynamy od krótkiego przypomnienia materiału z poprzedniej części zajęć i odtworzenia układu zbudowanego na poprzednich zajęciach.

Programowanie układu – 20 minut

Aby napisać kod obsługujący naszą klawiaturę, wykorzystamy istniejącą bibliotekę **Keypad.h**.

Biblioteki dołączamy poprzez dopisanie linii na początku naszego programu:

#include <Keypad.h>

Utwórzmy nowy szkic w Arduino IDE, spróbujmy dopisać tę bibliotekę na jego początku i zweryfikować czy kod się skompiluje. Niestety Arduino IDE wyświetli nam błąd:

„fatal error: Keypad.h: No such file or directory”

Biblioteka **Keypad.h** nie jest dołączona do naszego oprogramowania. By ją wykorzystać, musi ona zostać pobrana z zewnętrznego serwera. Robimy to w sposób następujący:

- w menu Arduino IDE klikamy **Sketch>Include Library>Manage Libraries**,
- wyświetli nam się okno **Library Manager**, w którym możemy dołączyć do naszego projektu różne dostępne biblioteki,
- w pasku wyszukiwania w prawym górnym rogu wpisujemy **Keypad.h**. Pokaże nam się kilka wyników – szukamy takiego, którego autorami są Mark Stanley i Alexander Brevig.
- Klikamy na przycisk **Install** w prawym dolnym rogu dodatku,
- po krótkim procesie instalacji biblioteka wyświetli się w **Library Manager** jako **Installed**.

Jeżeli teraz spróbujemy skompilować nasz szkic z zaimportowaną na początku biblioteką Keypad.h, kompilacja powinna przebiec bez błędów. Następnie dopisujemy resztę kodu zgodnie z poniższym przykładem:

```
#include <Keypad.h>
```

```
const byte ilosc_wierszy = 4; //cztery wiersze
```

```
const byte ilosc_kolumn = 4; //cztery kolumny
```

```
char klawisze[ilosc_wierszy][ilosc_kolumn] = {
```

```
  {'1','2','3'},
```

```
  {'4','5','6'},
```

```
  {'7','8','9'},
```

```
  {'#','0','*'}
```

```
};
```

```
byte pinyWierszy[ilosc_wierszy] = {5, 4, 3, 2}; //do których pinów są  
podłączone wiersze
```

```
byte pinyKolumn[ilosc_kolumn] = {9, 8, 7, 6}; //do których pinów sa  
podłączone kolumny
```

```
// Tworzymy teraz tzw. klasę Keypad z instacją "keypad" do której zostają
```

```
// przypisane piny i wartości przyporządkowane ich kombinacjom
```

```
Keypad keypad = Keypad( makeKeymap(klawisze), pinyWierszy,  
pinyKolumn, ilosc_wierszy, ilosc_kolumn );
```

```
void setup(){
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop(){
```

```
  char klawisz = keypad.getKey();
```

```
  if (klawisz != NO_KEY){
```

```
    Serial.println(klawisz);
```

```
  }
```

```
}
```

Po uruchomieniu Serial Monitor i naciśnięciu klawiszy, powinniśmy zobaczyć wyświetlające się w nim znaki które naciśniemy na klawiaturze.

Zadanie dodatkowe – 15 minut

Do wolnego pinu analogowego podłączyć brzęczyk (buzzer) zgodnie ze scenariuszem 15. *Buzzer – piezoelektryczny głośniczek*, oraz przeprogramować kod tak, by zrobić klawiaturę muzyczną, grającą różne dźwięki przy pomocy brzęczyka.

MOŻLIWE MODYFIKACJE DLA MŁODSZYCH KLAS:

Pracując z uczniami w młodszych klasach można wykorzystać zamiast Arduino IDE program mBlock. W przypadku zajęć z młodszymi dziećmi warto zwrócić uwagę na ewentualne problemy z dokładnym podłączaniem przewodów.

Klasy I-III mogą wykonać zadanie manualnie zasilając jedną kolumną z użyciem digitalWrite, i robić digitalRead na pinach do których podpięte są rzędy, żeby manualnie zmapować jakie wartości zasilania na której kolumnie zwracają stan wysoki na których pinach wierszy.

Klasy IV-VI mogą wykonać zadania dodatkowe jeżeli pozwoli na to czas i doświadczenie uczestników.

ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS ZAJĘĆ:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole buduje układ z wykorzystaniem Arduino, płytki stykowej, diody LED i oporników. Zadanie polega na zmianie wartości przypisanych do poszczególnych przycisków keypada, oraz objaśnieniu zasady działania rzędów i kolumn.

PIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH:

Kurs programowania Arduino Forbot:

<http://forbot.pl/blog/artykuly/programowanie/kurs-arduino-w-robotyce-1-wstep-id936>

Podstawowe informacje na temat prądu elektrycznego:

<http://forbot.pl/blog/artykuly/podstawy/podstawy-elektroniki-1-napiecie-prad-opor-zasilanie-id3947>

Jak działa płytka stykowa (prototypowa):

https://pl.wikipedia.org/wiki/P%C5%82ytka_prototypowa

Przykład odczytu z kolumn i wierszy z objaśnieniem

<https://www.youtube.com/watch?v=RGkIHFQf8-w>

Więcej o multipleksowaniu:

<https://pl.wikipedia.org/wiki/Multipleksowanie>

Scenariusz został opracowany na potrzeby projektu „MoboLab – roboty i tablety w Twojej szkole”. Celem projektu jest zwiększenie kompetencji informatycznych z zakresu programowania i wykorzystywania technologii mobilnych w uczeniu się, a także kreatywności, innowacyjności i umiejętności współpracy w zespole z wykorzystaniem TIK, uczniów / uczennic z (UCZ) z 6 szkół podnadgimnazjalnych i 4 gimnazjów Wołomina i Zielonki. Projekt dofinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego (Regionalny Program Operacyjny Województwa Mazowieckiego na lata 2014-2020, Oś Priorytetowa X. Edukacja dla rozwoju regionu, Działanie 10.1. Edukacja ogólna i przedszkolna, Poddziałanie 10.1.2. Edukacja ogólna w ramach ZIT).



Ten utwór jest dostępny na licencji [Creative Commons Uznanie autorstwa 4.0 Międzynarodowe](https://creativecommons.org/licenses/by/4.0/).