

Katedra podstaw elektroniki

Ćwiczenie:

Budowa pianina przy użyciu elementów optoelektronicznych.

1. Wstęp:

Celem ćwiczenia jest budowa pianina oraz zapoznanie się z podstawami języka C, który zostanie wykorzystany przy programowaniu go. Do sterowania tym pianinem zastosujemy mikrokontroler Arduino. Końcowym efektem powinna być „mini” klawiatura za pomocą której można będzie odgrywać określone dźwięki używając małej latarki lub lasera.

W ćwiczeniu będziemy wykorzystywać płytkę Arduino Leonardo, dlatego wielkość klawiatury będzie ograniczona.

2. Elementy, które zostaną użyte w ćwiczeniu oraz ich krótki opis:

- Arduino Leonardo

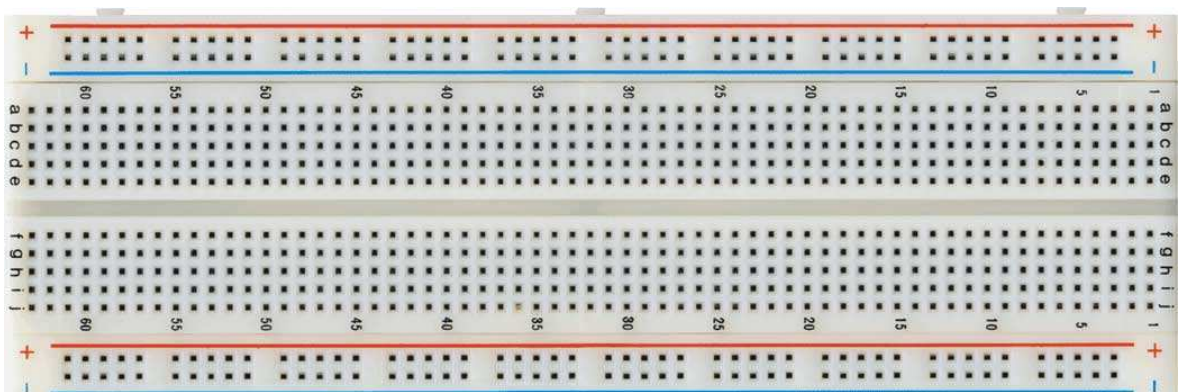
Mikrokontroler oparty na Atmega32u4 posiadający 20 wejść/wyjść, z czego 7 może być wykorzystane jako kanały PWM i 12 jako analogowe wejścia. Układ Atmega32u4 taktowany jest sygnałem o częstotliwości 16MHz, posiada 32 kB pamięci programu Flash, 1kB EEPROM oraz 2,5 kB pamięci operacyjnej SRAM. Aby zacząć wystarczy podłączyć go do komputera za pomocą kabla USB lub zasilać go baterią. Można także rozszerzać możliwości Arduino stosując nakładki (tzw. Arduino Shield).



Arduino Leonardo

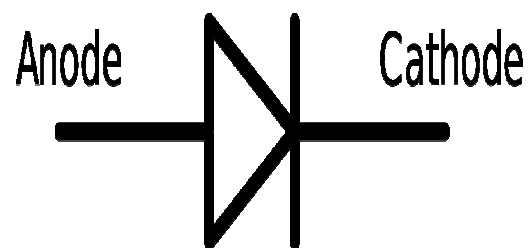
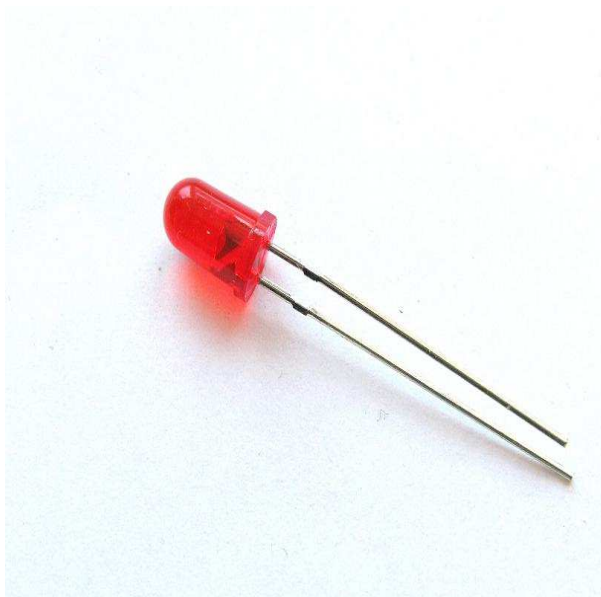
- płytki stykowej

Płytki o 830 otworach na której zostanie zbudowane pianino. Wykorzystując płytkę stykową w bardzo łatwy sposób można budować różne konstrukcje bez potrzeby wyciągania zakurzonej lutownicy. Plusiem jest możliwość jej wielokrotnego użytku.



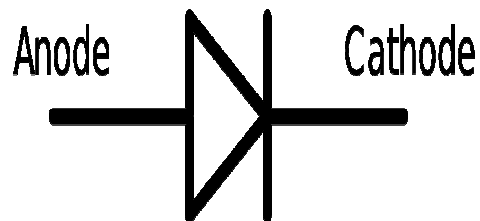
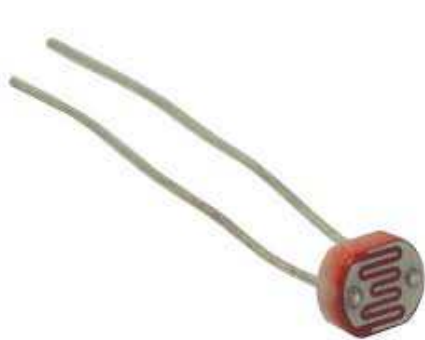
- dioda LED

Dioda świecąca LED jest elementem półprzewodnikowym, której działanie opiera się na zjawisku fizycznym zwanym elektroluminescencją. Pod wpływem przepływającego przez diodę prądu emituje ona światło widzialne lub niewidzialne jak podczerwone i ultrafioletowe. Po podłączeniu zasilania do diody, elektrony rekombinują z dziurami generując energię w postaci światła. W ćwiczeniu wykorzystane będą czerwone diody LED, jednak w celach wizualnych można wybrać różne kolory. Należy pamiętać, że różne diody mają różne napięcia przewodzenia.



- fotorezystory

Element półprzewodnikowy, który zmienia swoją rezystancję pod wpływem padającego na niego światła. Świecąc na niego laserem, będzie zmieniane napięcie na wejściu analogowym.



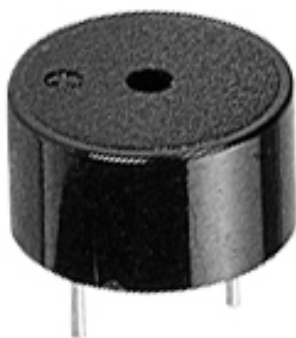
- rezystory

Rezystor zwany inaczej opornikiem. Rezystor służy do redukcji napięcia w obwodzie. Przy przepływie prądu przez opornik wytwarzane jest ciepło. W ćwiczeniu wykorzystane będą oporniki o rezystancji $2.2k\ \Omega$.



- buzzer

Sterowany elektrycznie sygnalizator akustyczny, który będzie pełnił rolę głośnika. W ćwiczeniu wykorzystany będzie buzzer 5V z wbudowanym generatorem.

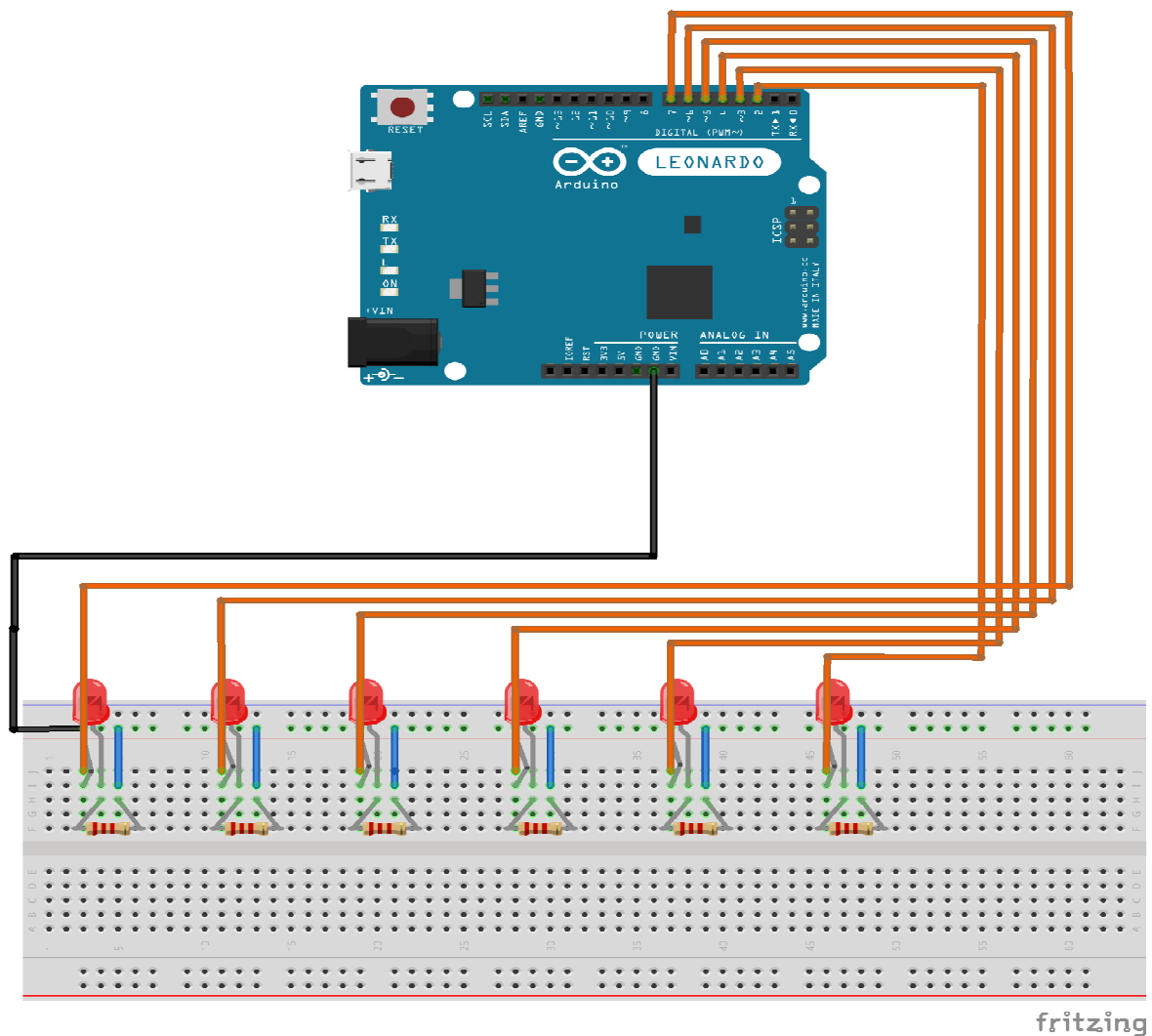


3. Montaż układu:

3.1. Na samym początku trzeba podłączyć diody w taki sposób, że mają być one sterowane przez wyjścia cyfrowe. Wiersz ze znakiem minus(-) na płytce stykowej podpiąć do pinu GND na płytce stykowej. Wpiąć diodę w płytkę stykową i podpiąć jej dłuższą nóżkę do wyjścia cyfrowego. Do drugiej nóżki podłączyć rezystor $2.2k\Omega$, natomiast rezystor połączyć do wiersza ze znakiem minus(-), który został wpięty wcześniej do gniazda GND.

Następujące czynności powtórzyć sześciokrotnie.

Układ powinien wyglądać jak ten na obrazku poniżej. Jeżeli wygląda inaczej wykonaj punkt 3.1. jeszcze raz.



fritzing

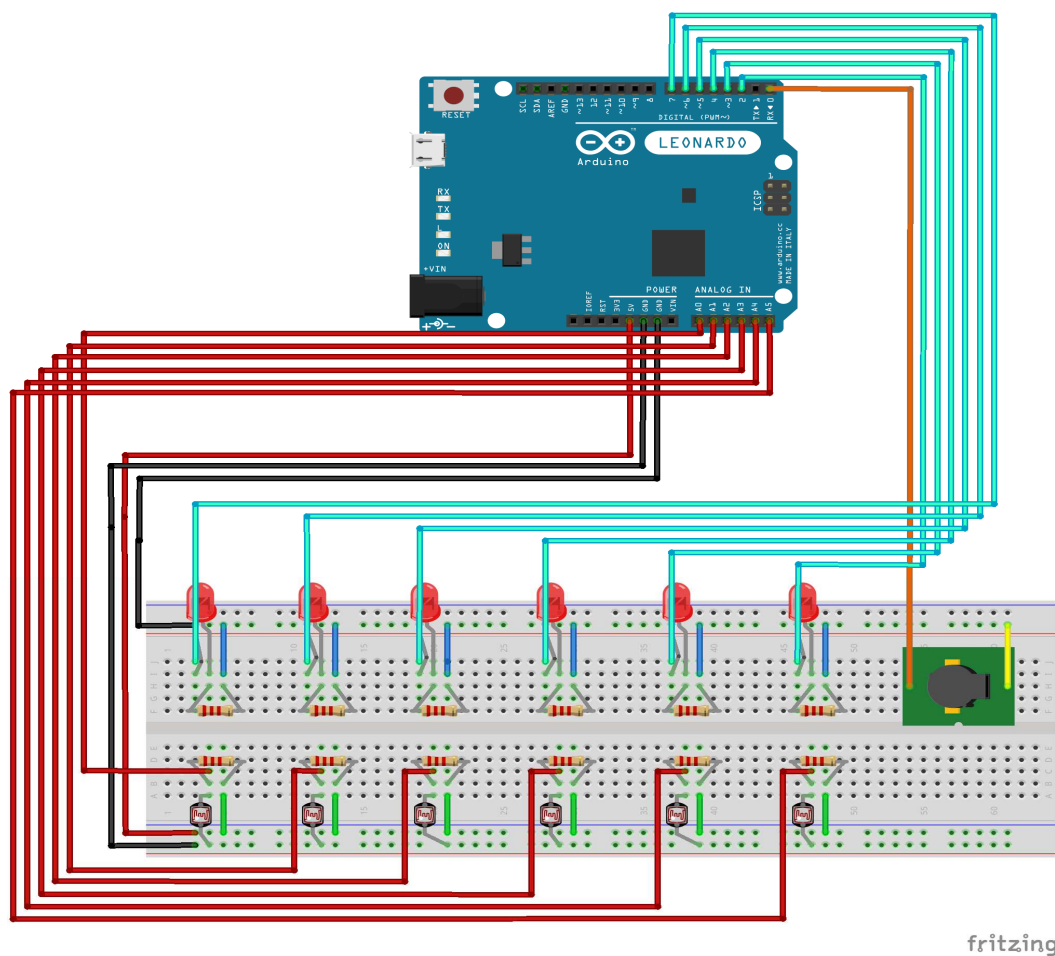
3.2. W drugim etapie trzeba podłączyć fotorezystory do wejść analogowych, które będą sprawdzały czy pada na nie światło.

Na początku trzeba podpiąć wiersz plus(+) na płytce stykowej do pinu 5V na Arduino oraz wiersz minus(-) do pinu GND. Fotorezystor jedną nóżką wpiąć do wiersza minus(-) na płytce stykowej. Do drugiej nóżki podpiąć rezystor 2.2k Ω , a rezystor połączyć z wierszem plus(+). Miejsce w którym łączy się rezystor z fotorezystorem wpiąć do wejścia analogowego na płytce Arduino. Czynności powtórzyć sześciokrotnie.

3.3. Ta część będzie najkrótsza, ponieważ trzeba wpiąć tylko jeden element – buzzer.

Plus brzęczyka wpiąć do wyjścia cyfrowego na płytce Arduino, a minus do wiersza minus(-) na płytce stykowej.

Po wykonaniu wszystkich etapów budowy układu powinien on wyglądać jak na obrazku poniżej. Jeśli wygląda inaczej skoryguj błędy.



4. Programowanie modu³u Arduino

Do programowania Arduino stosowany jest język C.

Program do lista instrukcji, które wykonują się w kolejności ich zapisania. W programie Arduino(sketch) znajdują się dwie funkcje, które muszą się zawsze znaleźć:

- void setup();
- void loop();

W funkcji setup() ustala się czy określone piny mają mieć charakter wejściowy czy wyjściowy.

Funkcja loop() zawiera instrukcje, które wykonują się w pętli podczas działania Arduino.

Oprócz funkcji setup() i loop() można samemu pisać własne funkcje lub wykorzystywać funkcje napisane przez kogoś innego(np. digitalWrite).

Oprócz funkcji istnieją zmienne, które przechowują informacje(np. int ledPin = 13;).

Przy wielu zmianach w programie pomaga skrócić czas tych zmian.

Każda zmienna przed użyciem musi zostać zadeklarowana(np. int delay = 500; lub bool swieci = false;).

Przy pisaniu programów wykorzystuje się polecenia takie jak :

```
- if(warunek)
{
    Instrukcje;
}
```

Instrukcje są wykonywane tylko wtedy, gdy warunek zostanie spełniony.

```
- for(int i=0;i<ile_razy;i++)
{
    Instrukcje;
}
```

Instrukcję będą wykonywane do momentu, gdy będzie spełniony warunek(i<ile_razy).

-while – polecenie podobne do polecenia for,

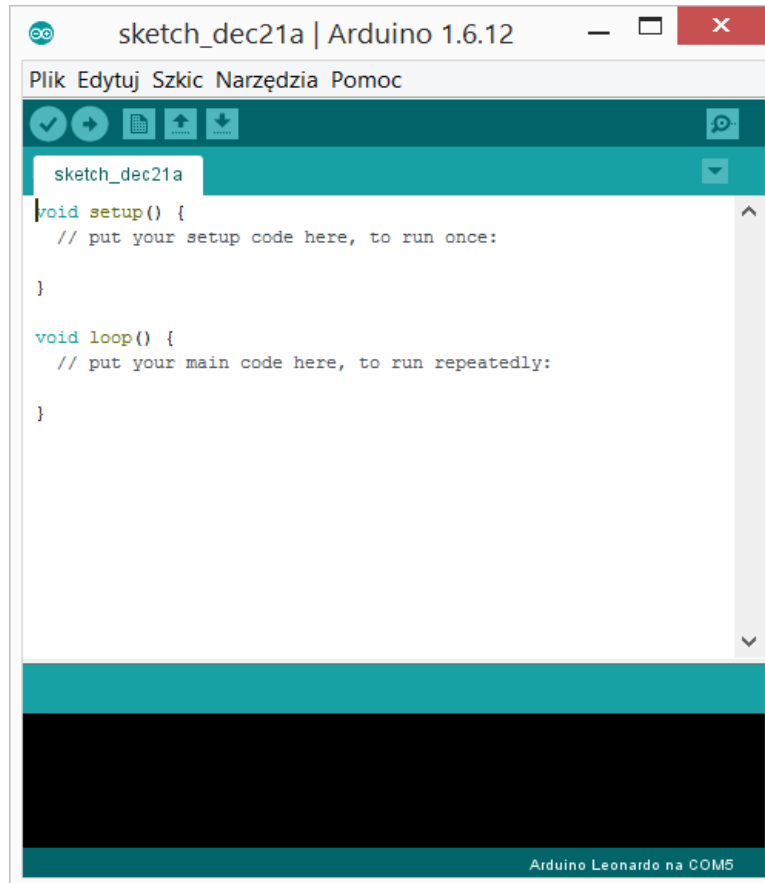
```
while(warunek)
{
    Instrukcje;
}
```

Instrukcję będą wykonywane do momentu, gdy będzie spełniony warunek.

5. Zaprogramowanie pianina przy użyciu języka C.

Przed wykonaniem tej części upewnij się, że poprawnie zmontowany został układ.

Do pisania programów na Arduino potrzebne jest środowisko Arduino IDE, które można bezpłatnie pobrać ze strony producenta. Po uruchomieniu utwórz nowy szkic (Plik -> Nowy). Środowisko powinno wyglądać tak jak na obrazku poniżej.



- 5.1. Pierwszym krokiem będzie zdefiniowanie częstotliwości dźwięków, jakie będą wygrywane na brzęczyku oraz uzupełnienie funkcji setup. Aby zmusić cię do większego zaangażowania, niektóre fragmenty kodu będą usunięte i trzeba będzie je samodzielnie napisać.
- Aby uzupełnić częstotliwości dźwięków można skorzystać z poniższej tabeli:

	Kontra	Wielka	Mała	Razkreślina	Dwukreślina	Trzykreślina	Czterokreślina
C	32,7	65,4	130,8	261,6	523,2	1046,5	2093,0
C#	34,6	69,3	138,6	277,2	554,4	1108,7	2217,4
D	36,7	73,4	146,8	293,7	587,3	1174,7	2349,3
D#	38,9	77,8	155,6	311,1	622,3	1244,5	2489,0
E	41,2	82,4	164,8	329,6	659,3	1318,5	2637,0
F	43,7	87,3	174,6	349,2	698,5	1396,9	2793,8
F#	46,2	92,5	185,0	370,0	740,0	1480,0	2959,9
G	49,0	98,0	196,0	392,0	784,0	1568,0	3136,0
G#	51,9	103,8	207,7	415,3	830,6	1661,2	3322,4
A	55,0	110,0	220,0	440,0	880,0	1760,0	3520,0
A#	58,3	116,5	233,1	466,2	932,3	1864,7	3729,3
H	61,7	123,5	246,9	493,9	987,8	1975,5	3951,1

Stosując dyrektywę preprocesora #define, wybierz określony dźwięk i zapisz jego częstotliwość wykorzystując kolumnę „Dwukreślina”.

Uwaga! Częstotliwości należy zapisać jako liczby całkowite. W funkcji setup należy określić czy piny do których są podpięte diody LED i buzzer mają pełnić funkcje INPUT czy OUTPUT. Dla wejść/wyjść analogowych nie jest to wymagane. Środowisko powinno wyglądać jak na obrazku

poniżej. W miejscu „//freq” powinna się znaleźć odpowiednia częstotliwość.

W pierwszym argumencie funkcji pinMode i digitalWrite powinien się znaleźć numer pinu, do którego podpięta jest dioda/buzzer.



```
sketch_dec21a$
#define C //freq
#define D //freq
#define E //freq
#define F //freq
#define G //freq
#define A //freq
void setup() {
  // put your setup code here, to run once:
  pinMode(dioda_1,OUTPUT);
  pinMode(diod1_2,OUTPUT);
  pinMode(dioda_3,OUTPUT);
  pinMode(dioda_4,OUTPUT);
  pinMode(dioda_5,OUTPUT);
  pinMode(dioda_6,OUTPUT);
  pinMode(buzzer,OUTPUT);
  digitalWrite(dioda_1,LOW);
  digitalWrite(dioda_2,LOW);
  digitalWrite(dioda_3,LOW);
  digitalWrite(dioda_4,LOW);
  digitalWrite(dioda_5,LOW);
  digitalWrite(dioda_6,LOW);
  digitalWrite(buzzer,LOW);
}

void loop() {
```

- 5.2. Następnym krokiem będzie zdefiniowanie własnej funkcji, która będzie miała za zadanie sprawdzić czy na fotorezystor pada światło i włączyć diodę przypisaną określonemu fotorezystorowi oraz wygrać na brzęczyku określony dźwięk, lub wyłączyć diodę. Następnie w funkcji loop wywołać tę funkcję dla wszystkich fotorezystorów. Funkcja będzie miała nazwę „zmiana” i będzie przyjmować trzy argumenty:

- numer pinu na którym ma sprawdzić fotorezystor,
- numer pinu który ma zaświecić diodę,
- częstotliwość dźwięku

Wewnątrz funkcji zostanie użyte polecenie „if”, zmienna globalna „napięcie”, funkcja analogRead, funkcja digitalWrite oraz funkcja tone.

Zmienna „napięcie” ma przypisaną wartość, którą wykorzystuje się przy sprawdzaniu warunku świecenia światła na fotorezystor.

Uwaga! Wartość dobrać tak, aby diody nie świeciły się przy zwykłym oświetleniu w pomieszczeniu.

Funkcja analogRead ma za zadanie czytać napięcie jakie jest podawane na pin podany jako pierwszy argument.

Funkcja digitalWrite ma za zadanie włączyć lub wyłączyć napięcie na określonym pinie, który jest przyjmowany jako pierwszy argument. W drugim argumencie podawane jest napięcie jakie ma wystać(HIGH – 5V, LOW – 0V).

Funkcja tone ma za zadanie włączyć dźwięk na urządzeniu podłączonym do pinu który podany jest jako pierwszy argument. Drugim argumentem jest częstotliwość dźwięku, a trzecim długość dźwięku w milisekundach. Ponieważ instrukcje będą wykonywane w pętli w trzecim argumencie należy podać wartość nie większą niż 120 ms.

Ostatecznie funkcja loop i zmiana ma postać:

```
void loop() {  
    // put your main code here, to run repeatedly:  
  
    zmiana(0,7,C1);  
    zmiana(1,6,D1);  
    zmiana(2,5,E1);  
    zmiana(3,4,F1);  
    zmiana(4,3,G1);  
    zmiana(5,2,A1);  
}  
void zmiana(int wejscie,int wyjscie,int freq)  
{  
    if(analogRead(wejscie)<napiecie)  
    {  
        digitalWrite(wyjscie,HIGH);  
        tone(0,freq,90);  
    }  
    else  
    {  
        digitalWrite(wyjscie,LOW);  
    }  
}
```

6. Wgranie szkicu na płytkę Arduino i sprawdzenie wyniku pracy.

Aby wgrać szkic należy podłączyć płytkę Arduino z komputerem za pomocą kabla USB. W pasku narzędzi wybrać „Narzędzia->Płytkę->Arduino Leonardo” oraz „Narzędzia->Port” i wybrać port, do którego podłączona jest płytkę. Następnie kliknąć „Wgraj” i poczekać aż wyświetli się komunikat taki jak na obrazku poniżej.

```
Ładowanie zakończone.  
Zmienne globalne używają 181 bajtów (7%) pamięci dynamicznej, pozd
```

Jeśli wystąpi błąd przy kompilacji należy poprawić błędy w programie. Natomiast, gdy błąd wystąpi podczas wgrywania programu sprawdź czy komputer wykrywa płytke.

Sprawdź wyniki swojej pracy. Jeśli świecąc laserem po fotorezystorze słychać dźwięk i dioda zapala się to ćwiczenie zostało wykonane poprawnie.

