

## **PROGRAMOWANIE I ARDUINO W SZKOLE**

*Agnieszka Szymczak*  
*Szkoła Podstawowa w Wielkim Rychnowie*  
*Michał Szymczak*  
*Toruński Ośrodek Doradztwa Metodycznego i Doskonalenia Nauczycieli*  
*e-mail szymczak@abs.umk.pl*

*Abstract, Arduino is a modern programming platform. You can use it in classes at school to familiarize your students with operation rules of electronic devices and the basics of programming. To design systems student has to use prior knowledge of physics, technology and ICT. Arduino teaches creativity during designing systems and programming them.*

### **1. Wstęp**

XXI wiek oferuje człowiekowi dostęp do nowoczesnych technologii we wszystkich dziedzinach życia. Urządzenia, z których na co dzień korzystamy wyposażone są w ogromną ilość układów elektronicznych. Urządzenia te należy uruchomić, skonfigurować, a czasami nawet oprogramować. Dlatego na współczesnej szkole ciąży obowiązek wykształcenia w młodych pokoleniach podstaw działań tych urządzeń, a w konsekwencji skonfigurowania ich według własnych potrzeb i oczekiwań.. W konsekwencji pojawiania się nowoczesnych urządzeń elektronicznych nowe pokolenia powinny poznać na wszystkich etapach edukacyjnych podstawy działania urządzeń elektronicznych, jak również podstawy programowania. Według autorów podstawy programowej należą rozumieć to „nie tylko jako proces tworzenia programu w języku programowania, ale także proces związany z podejściem do rozwiązywania problemu: od specyfikacji problemu, przez znalezienie i opracowanie rozwiązania, do zaprogramowania rozwiązania, przetestowania jego poprawności i ewentualnej korekty przy użyciu odpowiednio dobranej aplikacji lub języka programowania.”

Jak zauważyli twórcy podstawy programowej programowanie to nie tylko proces tworzenia programu, ale procesy związane z odpowiednim podejściem do zadanego problemu, który może dotyczyć różnych dziedzin życia począwszy od planowania budżetu domowego, poprzez oprogramowanie urządzeń elektronicznych (programowanie urządzeń inteligentnego domu), a skończywszy na zadaniach zawodowych.

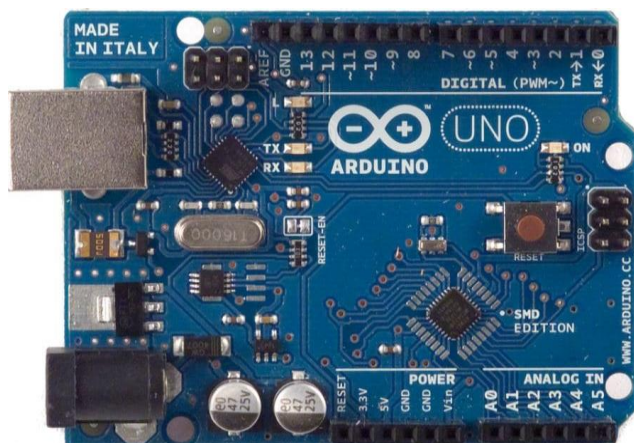
Poszukując narzędzi, które w sposób praktyczny i ciekaw przedstawią zagadnienia związane z projektowaniem układów elektronicznych, a po ich skonstruowaniu umożliwią ich oprogramowanie natrafiłem projekt platformy Arduino. Logo projektu przedstawiono na rysunku 1.



Rysunek 1 Logo projektu Arduino

## 2. Projekt Arduino

Projekt Arduino to platforma programistyczna, w której skład wchodzi: płytka drukowana oraz środowisko programistyczne. Projekt ten powstał w 2005 r. z inicjatywy profesora Massimo Banzii w celu zbudowania urządzenia kontrolującego studentki projekty jako alternatywa dla innych dostępnych systemów prototypowania. Zarówno sama płytka Arduino, jak również środowisko programistyczne oparte jest na otwartej licencji, co pozwala różnym producentom stworzyć własne rozwiązania inspirowane na oryginalnym Arduino rozprowadzanych pod innymi nazwami. W związku z tym na rynku można znaleźć wiele różnych rozwiązań zgodnych z Arduino. Na rysunku 2 przedstawiono oryginalną płytkę Arduino UNO, a na rysunku 3 jego klon.

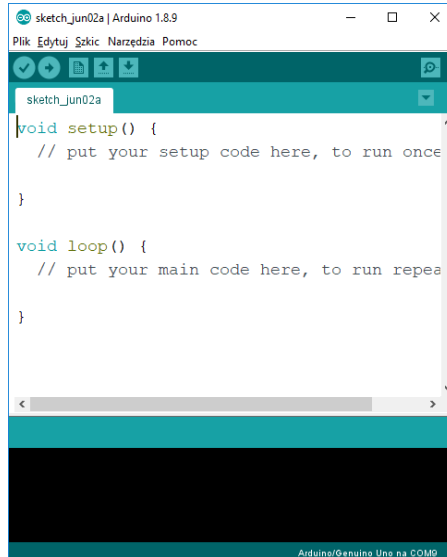


Rysunek 2 Arduino UNO



**Rysunek 3** Klon platformy Arduino

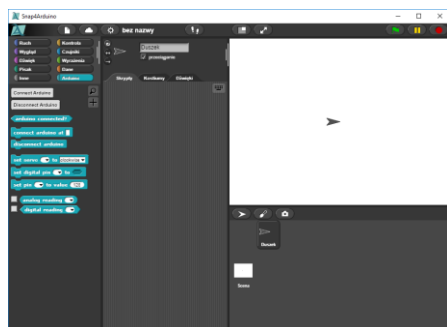
Nieodzownym elementem platformy jest środowisko programistyczne (rysunek 4), które pozwala oprogramować układ w języku programowania bazującym na C++. Oprogramowanie można pobrać ze strony <https://www.arduino.cc/>. Jest ono dostępne dla systemu Windows, Linux i Mac OS.



**Rysunek 4** Program Arduino IDE

Atutem projektu jest możliwość wykorzystania oprogramowania do wizualnego programowania Arduino. Jednym z nich jest aplikacja Snap4Arduino dostępnej na

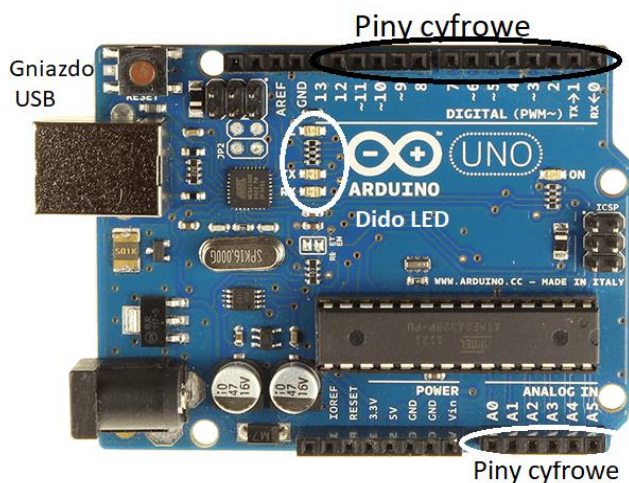
stronie <http://snap4arduino.rocks/>. Również dobrym narzędziem rozpoczęcia przygody z Arduino jest platforma Tinkercad dostępna pod adresem <https://www.tinkercad.com/>.



Rysunek 5 Środowisko Snap4Arduino

### 3. Budowa Arduino UNO

Płyta Arduino jest oparta na projekcie Open Hardware, w związku z tym mogą wystąpić różnice w budowie platformy. Dlatego też budowa zostanie omówiona na podstawie płyty Arduino UNO (rysunek 6). Działanie płyty Arduino oparte jest na 8 bitowym mikrokontrolerze firmy Atmel, AVR ATmega328 pracującego z częstotliwością 16 MHz. Po bokach płytki umieszczone zostały złącza służące do wyprowadzenia najważniejszych sygnałów (14 programowalnych cyfrowych wejść/wyjść numerowanych od 0 do 13, 6 cyfrowych analogowych wejść numerowanych od A0 do A5,



Rysunek 6 Budowa Arduino

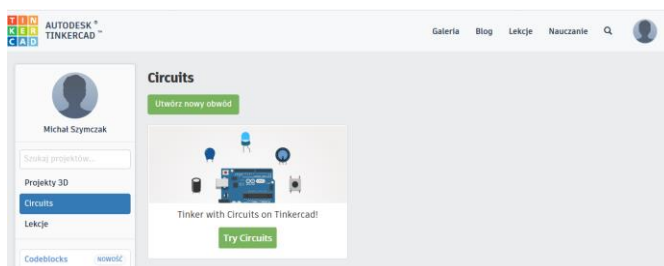
## 4. Arduino na start

Pierwsze kroki z platformą Arduino bez wykorzystania fizycznego urządzenia można rozpocząć korzystając z platformy <https://www.tinkercad.com/> (rysunek 7). Rejestracja jest darmowa. Projektowanie z wykorzystaniem platformy tinkercad nie powinno sprawiać trudności uczniom klas 7 i 8.



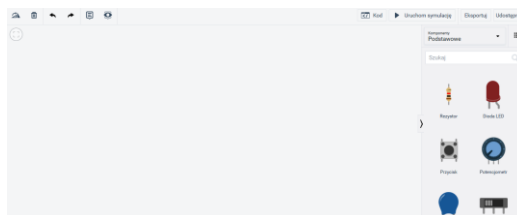
Rysunek 7 Strona projektu

Po utworzeniu konta i zalogowaniu się pojawia się strona zawierające przykładowe projekty. Z lewej strony od loginem wybieramy link Circuits (rysunek 8).



Rysunek 8 Okno projektów układów elektronicznych

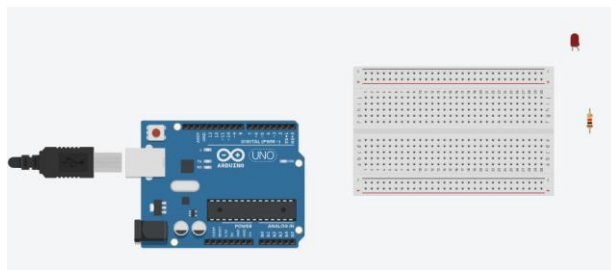
Abu utworzyć nowy projekt wybieramy przycisk Utwórz nowy obwód. Uruchomione zostanie nowe okno, w którym po prawej stronie znajdują się elementy elektroniczne, a główną część okna zajmuje obszar projektowy (rysunek 10).



Rysunek 9 Okno projektu układu elektronicznego

Pierwszy projekt, jaki utworzymy, będzie to migająca dioda. W polu roboczym umieszczamy (rysunek 8):

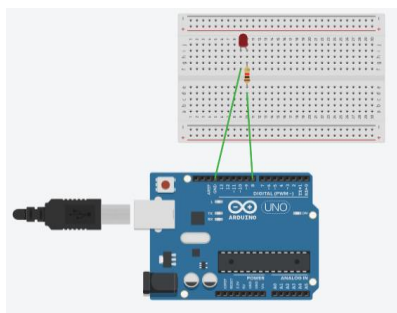
- płytke Arduino Uno
- płytke prototypową
- diodę LED
- rezystor (dla uproszczenia przyjmijmy, że wartość rezystora powinna wynosić  $180\ \Omega$ )



**Rysunek 10** Elementy projektu

Łączenia elementów dokonujemy za pomocą przewodów. W sposób pokazany na rysunku. Należy pamiętać o zachowaniu biegunowości diody LED.

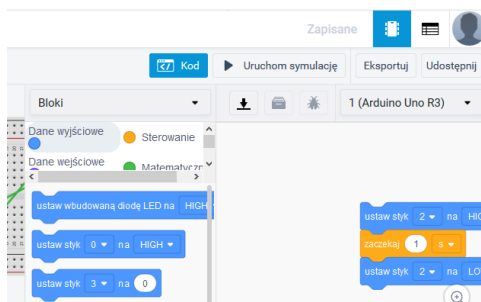
Krótsze wyprowadzenie diody LED (katoda) podłączamy do wtyku oznaczonym GND, a dłuższe wyprowadzenie diody do rezystora. Natomiast rezystor do pinu oznaczonego numerem 8 (rysunek 11).



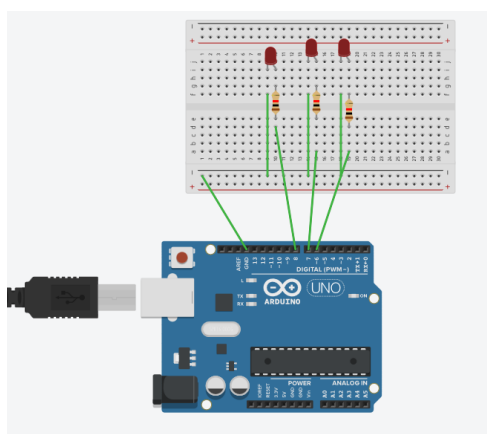
**Rysunek 11** Gotowy projekt

Programowanie układu odbywa się poprzez wybranie przycisku KOD. Program tworzymy za pomocą bloków (podobnie jak w Skretchu), ale dodatkowo możemy podejrzeć program w wersji C++ gotowy do wgrania na płytkę. Program ten nosi nazwę Sketch (rysunek 12).

Stwórzmy nowy projekt pokazujący zmianę światła na sygnalizatorze drogowym zgodnie ze schematem (rysunek 13).



Rysunek 12 Utworzenie programu do projektu



Rysunek 13 Elementy do projektu

W polu roboczym umieszczamy:

- płytkę Arduino Uno
- płytkę prototypową
- diody LED
- rezystory

Przykładowe rozwiązanie dla jednego sygnalizatora przedstawia się następująco (rysunek 14).

Program gotowy do wgrania do Arduino ma postać:

```
void setup(){
  pinMode(1, OUTPUT);
  pinMode(0, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
```

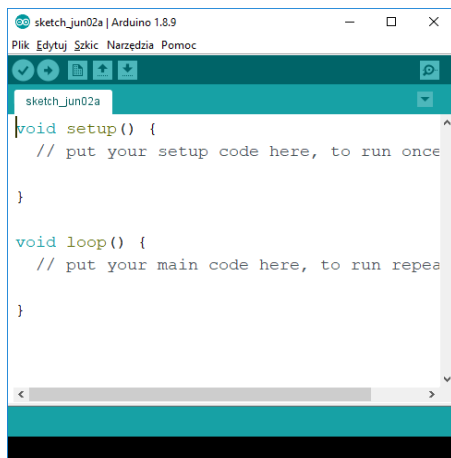
Rysunek 14 Program sygnalizatora świetlnego



```
pinMode(7, OUTPUT);
}
void loop(){
  digitalWrite(1, LOW);
  digitalWrite(0, HIGH);
  delay(2000); // Wait for 2000 millisecond(s)
  digitalWrite(1, HIGH);
  delay(2000); // Wait for 2000 millisecond(s)
  digitalWrite(0, LOW);
  digitalWrite(1, LOW);
  digitalWrite(2, HIGH);
  delay(2000); // Wait for 2000 millisecond(s)
  digitalWrite(2, LOW);
  digitalWrite(1, HIGH);
  delay(2000); // Wait for 2000 millisecond(s)
```

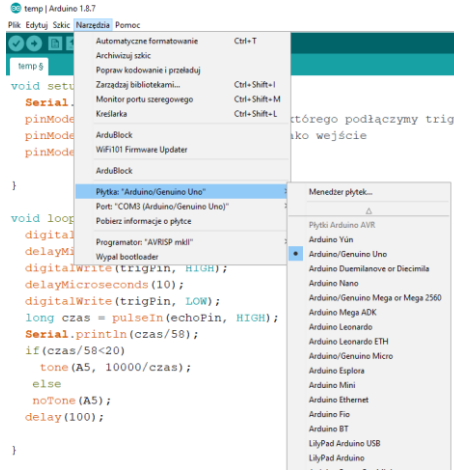
## 5. Arduino IDE

Projektowanie i programowanie platformy Arduino za pomocą narzędzi projektowych można zakończyć, w momencie, kiedy użytkownicy potrafią wykonywać projekty. W celu wgrania oprogramowania do płytki Arduino należy zainstalować pobrane ze strony <https://www.arduino.cc/> oprogramowanie Arduino IDE. Proces instalacji przebiega bardzo sprawnie. Po zainstalowaniu oprogramowania uruchamiamy oprogramowanie (rysunek 15). Należy wybrać odpowiednią płytkę i port, pod który jest ona podłączona (rysunek 16).



Rysunek 15 Program Arduino IDE





Rysunek 16 Ustawianie parametrów oprogramowania

## 6. Tworzenie programu w Arduino IDE

Zasadniczo program składa się z dwóch funkcji:

```
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Pierwsza z tych funkcji void setup() uruchamiana jest tylko raz podczas uruchomienia programu na platformie Arduino. Druga funkcja void loop() jest uruchamiana w nieskończonej pętli.

Przykładowy program, który będzie zapalał i gasił diodę podłączoną zgodnie z rysunkiem 17 przedstawiono poniżej.

```
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  digitalWrite(8, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(8, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Rysunek 17 Przykładowy program

Wygrzywanie jego do płytki odbywa się poprzez przycisk strzałki skierowanej w prawo (rysunek 18).



Rysunek 18 Menu oprogramowania Arduino

## Literatura

1. Monk S., *Arduino dla początkujących. Podstawy i szkice. Wydanie II*, Helion, Gliwice 2018.
2. Evans M., Noble J., Hochenbaum J., *Arduino w akcji*, Helion, Gliwice 2014.
3. Kurs Arduino, <https://forbot.pl/blog/kurs-arduino-podstawy-programowania-spis-tresci-kursu-id5290>, ostatni dostęp 1.06.2019 roku.