

Komputery jednokładowe

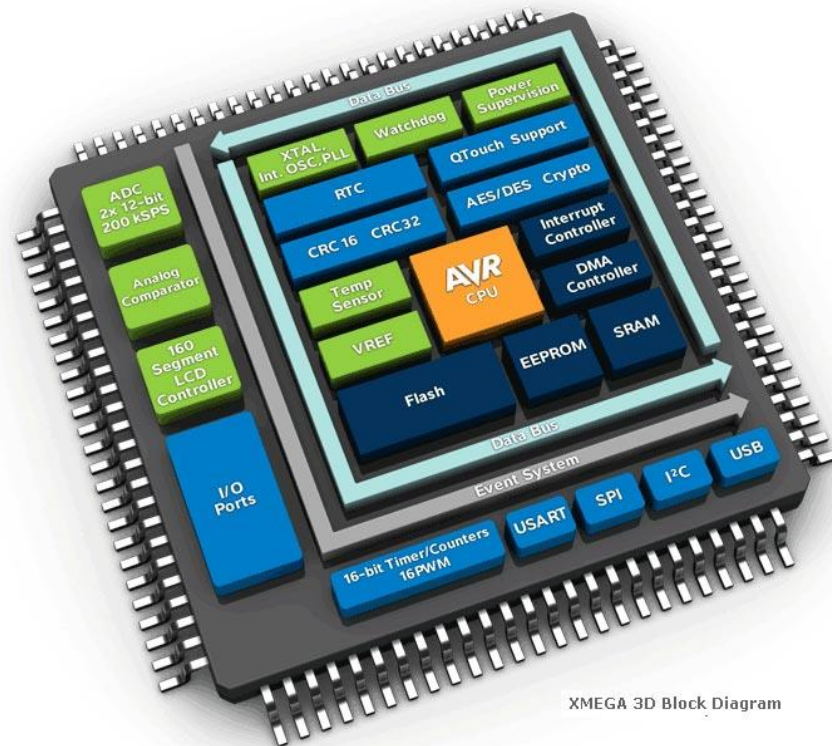
Kamil Cecherz



Definicja

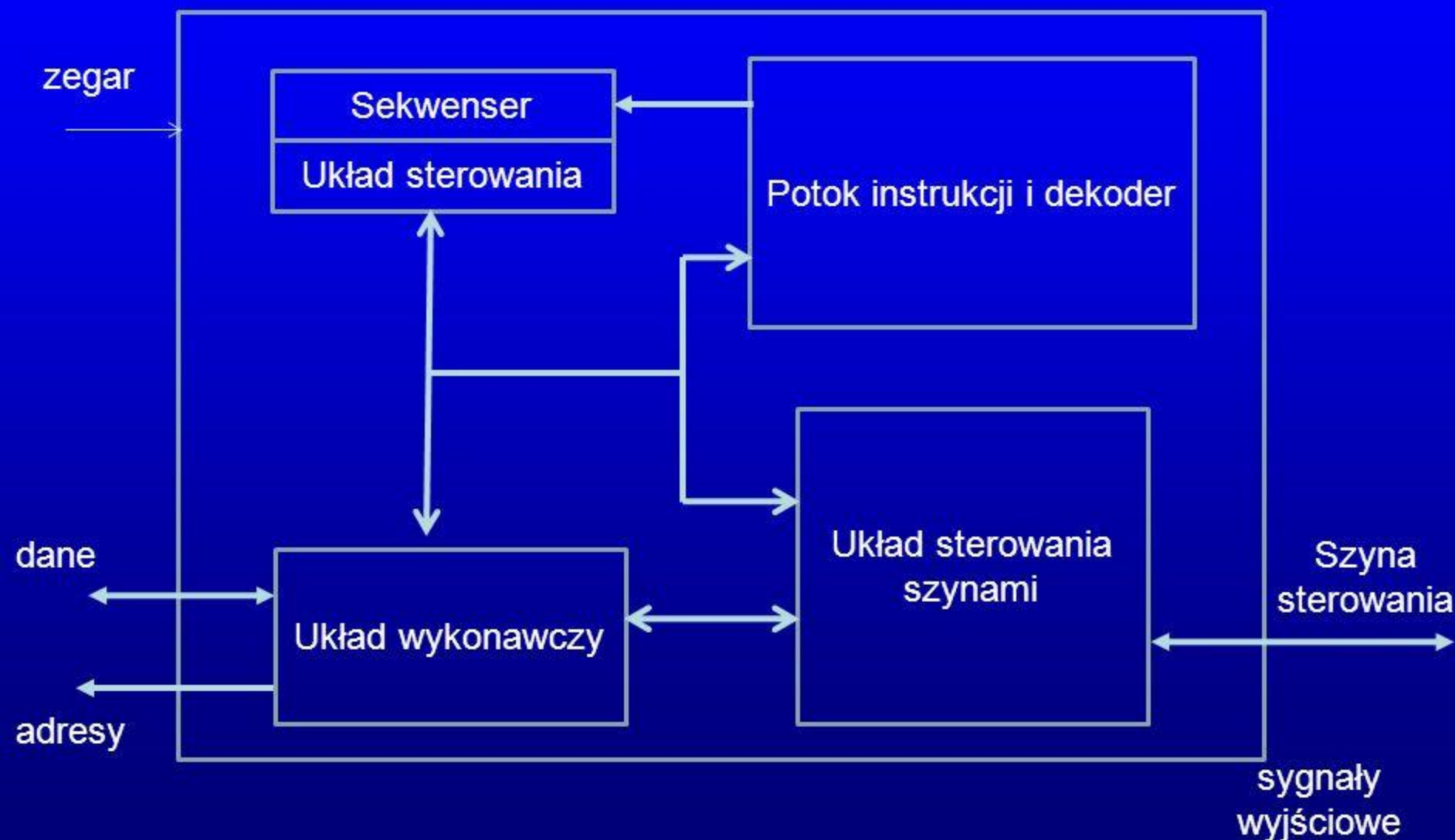
mikrokontroler

- Scalony system mikroprocesorowy, zrealizowany w postaci pojedynczego układu scalonego zawierającego jednostkę centralną (CPU), pamięć RAM oraz rozbudowane układy wejścia-wyjścia i na ogół pamięć programu jako FRAM, MRAM, ROM lub Flash.
- Określenie mikrokontroler pochodzi od głównego obszaru jego zastosowań, jakim jest sterowanie urządzeniami elektronicznymi, takimi jak: urządzenia biurowe, urządzenia medyczne (w tym implanty), zdalnego sterowania, elektronarzędzia, systemy sterowania silnikami samochodowymi, a nawet zabawki i inne systemy wbudowane.



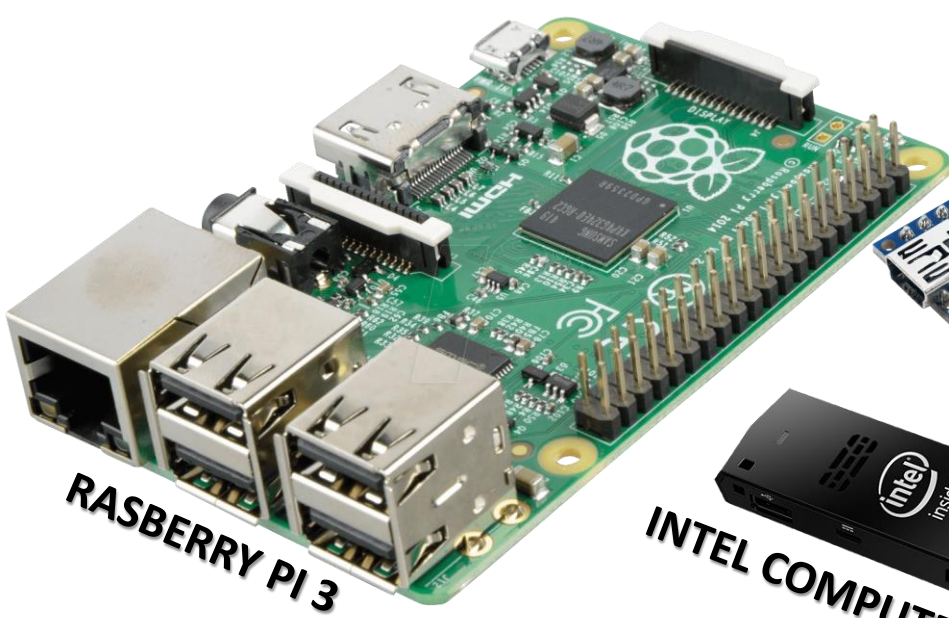
XMEGA 3D Block Diagram

Budowa typowego mikrokontrolera



Moduł SOM

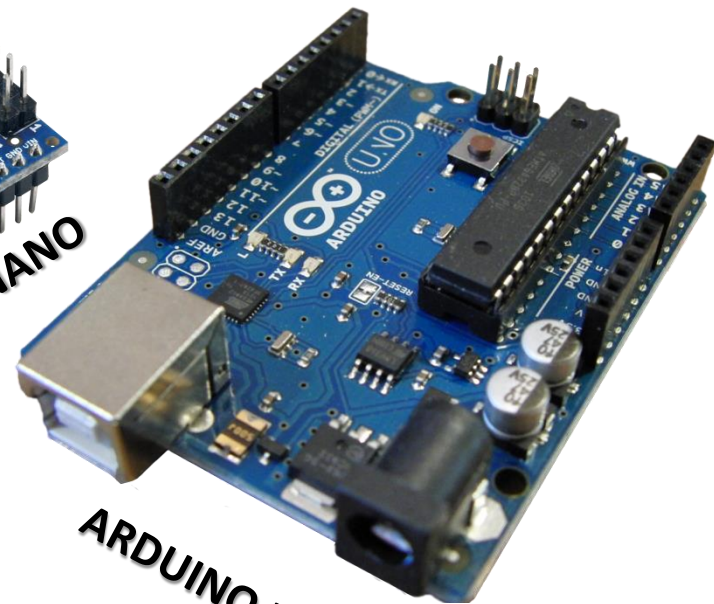
- Określany często także jako COM (Computer on Module) to niewielka płytką drukowana zawierająca mikroprocesor, układy IO, a także pamięć Flash i RAM, układy komunikacji bezprzewodowej Bluetooth i Wi-Fi, PHY do Ethernetu, układy audio i do zarządzania zasilaniem, a nawet kontroler ekranu dotykowego. Konstrukcja pozwala na uruchomienie systemu operacyjnego takiego jak Linux, Android czy Windows.



RASBERRY PI 3



ARDUINO NANO



ARDUINO UNO



INTEL COMPUTE STICKS



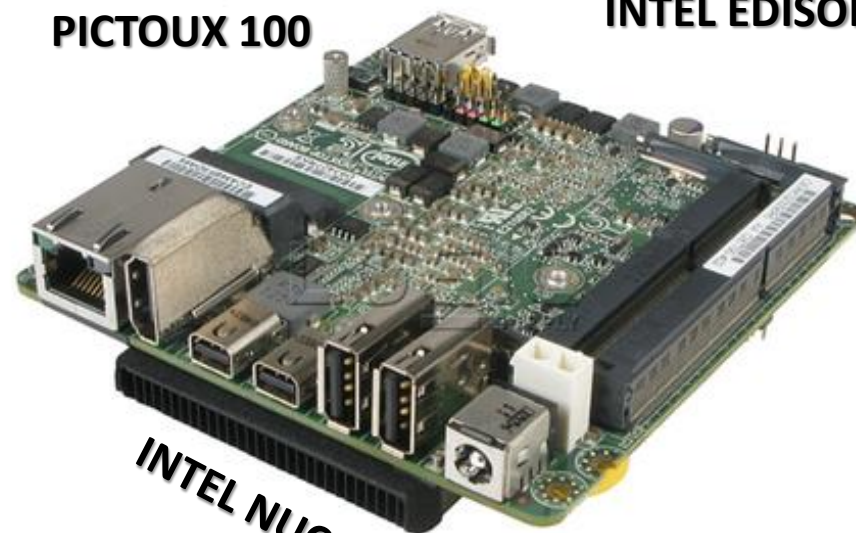
PICTOUX 100



INTEL EDISON

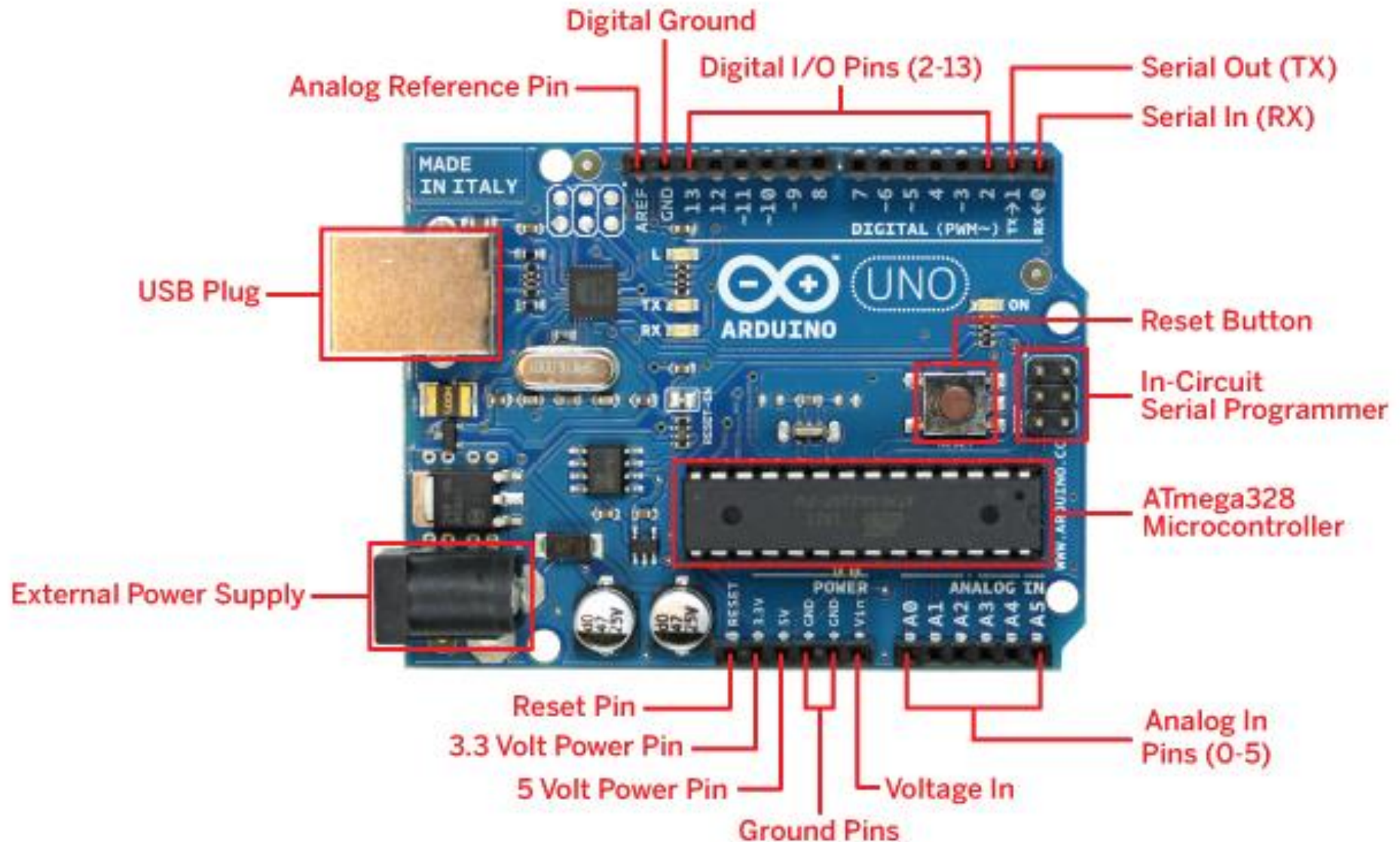


MAC MINI

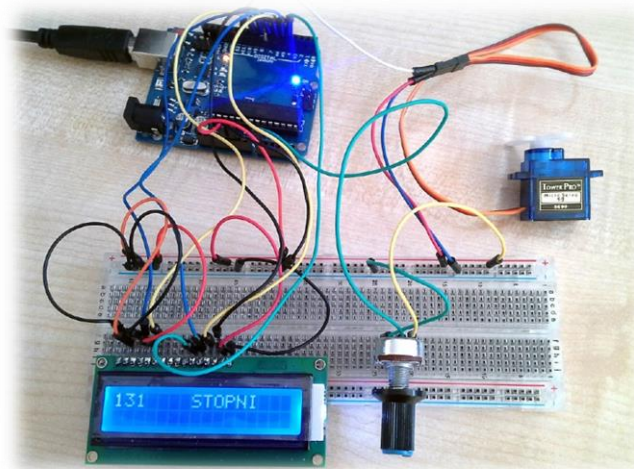
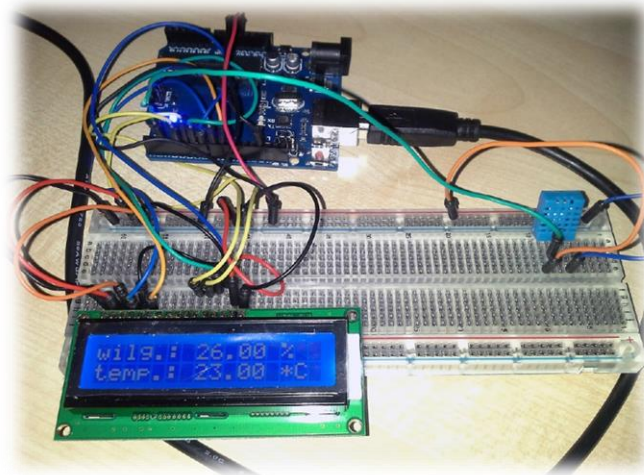
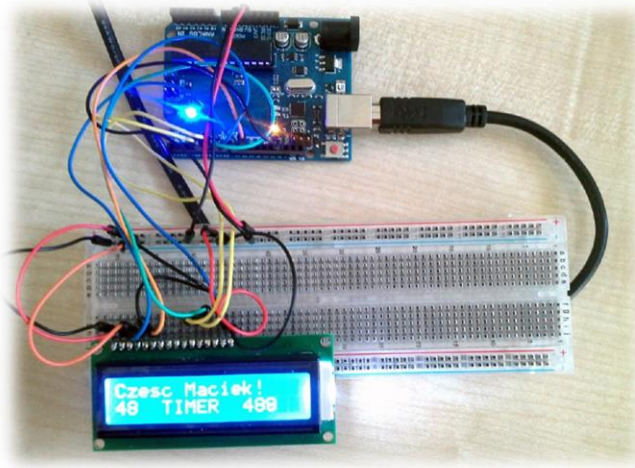


INTEL NUC

OPIS ZAWARTOŚCI PŁYTKI ARDUINO UNO



Przykłady zastosowania Arduino



Programowanie Arduino



- **Arduino IDE** jest wieloplatformową aplikacją napisaną w języku Java, wydzieloną z IDE przygotowanego dla języka Processing i projektu **Wiring**.
- Środowisko jest zaprojektowane w taki sposób, aby było przyjazne dla hobbystów i osób niezajmujących się tworzeniem oprogramowania. IDE zawiera edytor kodu z takimi funkcjami, jak podświetlanie składni czy automatyczne wcięcia w kodzie, oraz pozwala na kompilację i upload programu do płyty Arduino.
- Zazwyczaj nie ma potrzeby dodatkowej edycji plików Makefile lub uruchamiania programów z linii poleceń. Standardowo IDE Arduino zawiera bibliotekę C/C++ o nazwie "Wiring,, dzięki czemu wykonywanie podstawowych operacji wejścia/wyjścia staje się znacznie łatwiejsze. Programy dla Arduino są napisane głównie w języku **C/C++**, jednak użytkownicy muszą zdefiniować jedynie dwie funkcje, aby otrzymać gotowy do uruchomienia program:
 - ❑ **setup()** – funkcja wykonywana raz, na początku działania programu, wykorzystywana najczęściej do ładowania ustawień,
 - ❑ **loop()** – funkcja wywoływana wielokrotnie, przez cały okres działania programu, czyli do czasu odłączenia zasilania od układu.

PRZYKŁADOWE PROGRAMY

```
#include "WProgram.h"
#define LED_PIN 13

void setup () {
    pinMode (LED_PIN, OUTPUT);    // ustawienie pinu 13 jako cyfrowego wyjścia
}

void loop () {
    digitalWrite (LED_PIN, HIGH); // włączenie diody LED
    delay (1000);                 // oczekiwanie sekundy (1000 milisekund)
    digitalWrite (LED_PIN, LOW);  // wyłączenie diody LED
    delay (1000);                 // oczekiwanie sekundy
}

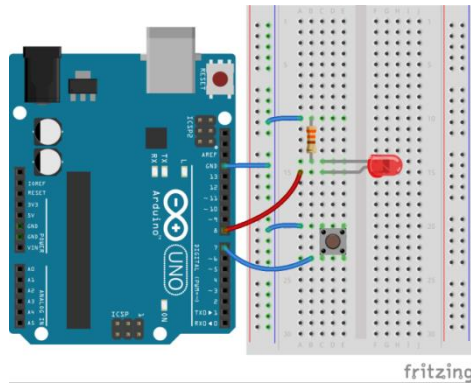
int main(void)
{
    init();

    setup();

    for (;;)
        loop();

    return 0;
}
```

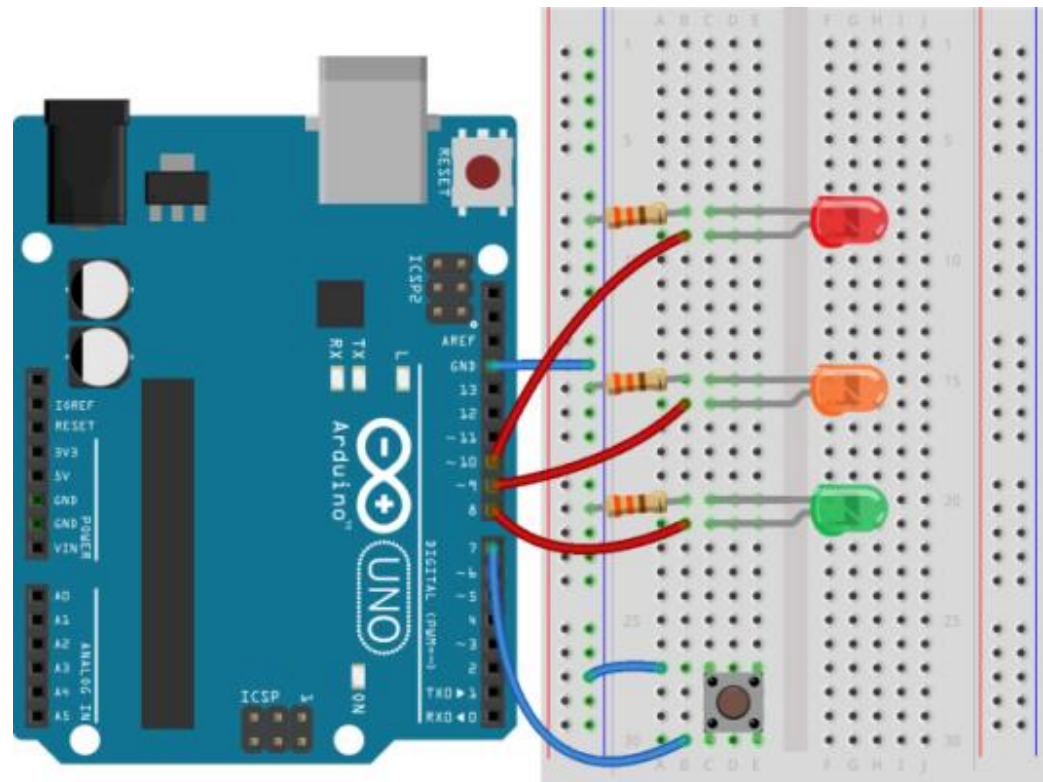
PRZYKŁADOWE PROGRAMY



```
void setup() {  
  pinMode(8, OUTPUT); //Dioda jako wyjście  
  pinMode(7, INPUT_PULLUP); //Przycisk jako wejście  
  digitalWrite(8, LOW); //Wyłączenie diody  
}  
  
void loop()  
{  
  if (digitalRead(7) == LOW) { //Jeśli przycisk wciśnięty  
    digitalWrite(8, HIGH); //Włącz diodę  
  } else { //Jeśli warunek nie został spełniony (przycisk nie jest wciśnięty)  
    digitalWrite(8, LOW); //Wyłącz diodę  
  }  
}
```

PRZYKŁADOWE PROGRAMY

```
void setup() {  
  pinMode(10, OUTPUT); //Dioda czerwona  
  pinMode(9, OUTPUT); //Dioda żółta  
  pinMode(8, OUTPUT); //Dioda zielona  
  
  pinMode(7, INPUT_PULLUP); //Przycisk  
  
  digitalWrite(10, LOW); //Wyłączenie diod  
  digitalWrite(9, LOW);  
  digitalWrite(8, LOW);  
}  
  
void loop()  
{  
  digitalWrite(10, LOW); //Czerwona  
  digitalWrite(9, LOW); //Pomarańczowa  
  digitalWrite(8, HIGH); //Zielona  
  
  delay(1000); //Czekamy 1 sekundę  
  
  digitalWrite(10, LOW); //Czerwona  
  digitalWrite(9, HIGH); //Pomarańczowa  
  digitalWrite(8, LOW); //Zielona  
  
  delay(1000); //Czekamy 1 sekundę  
  
  digitalWrite(10, HIGH); //Czerwona  
  digitalWrite(9, LOW); //Pomarańczowa  
  digitalWrite(8, LOW); //Zielona  
  
  delay(1000); //Czekamy 1 sekundę  
  
  digitalWrite(10, HIGH); //Czerwona  
  digitalWrite(9, HIGH); //Pomarańczowa  
  digitalWrite(8, LOW); //Zielona  
  
  delay(1000); //Czekamy 1 sekundę  
}
```



PRZYKŁADOWE PROGRAMY

```
void setup() {
  pinMode(10, OUTPUT); //Dioda czerwona
  pinMode(9, OUTPUT); //Dioda żółta
  pinMode(8, OUTPUT); //Dioda zielona

  pinMode(7, INPUT_PULLUP); //Przycisk

  digitalWrite(10, LOW); //Wyłączenie diod
  digitalWrite(9, LOW);
  digitalWrite(8, LOW);
}

void loop()
{
  digitalWrite(10, LOW); //Czerwona
  digitalWrite(9, LOW); //Pomarańczowa
  digitalWrite(8, HIGH); //Zielona

  delay(1000); //Zatrzymujemy program przed wejściem do pętli na 1 sekundę
  while (digitalRead(7) == HIGH) {} //Jeśli przycisk jest wciśnięty

  digitalWrite(10, LOW); //Czerwona
  digitalWrite(9, HIGH); //Pomarańczowa
  digitalWrite(8, LOW); //Zielona

  delay(1000); //Zatrzymujemy program przed wejściem do pętli na 1 sekundę
  while (digitalRead(7) == HIGH) {} //Jeśli przycisk jest wciśnięty

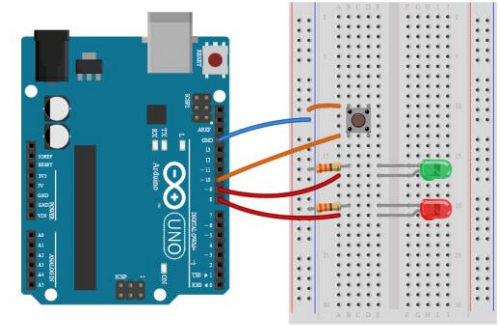
  digitalWrite(10, HIGH); //Czerwona
  digitalWrite(9, LOW); //Pomarańczowa
  digitalWrite(8, LOW); //Zielona

  delay(1000); //Zatrzymujemy program przed wejściem do pętli na 1 sekundę
  while (digitalRead(7) == HIGH) {} //Jeśli przycisk jest wciśnięty

  digitalWrite(10, HIGH); //Czerwona
  digitalWrite(9, HIGH); //Pomarańczowa
  digitalWrite(8, LOW); //Zielona

  delay(1000); //Zatrzymujemy program przed wejściem do pętli na 1 sekundę
  while (digitalRead(7) == HIGH) {} //Jeśli przycisk jest wciśnięty
}
```

PRZYKŁADOWE PROGRAMY



```
#define zielona 8
#define czerwona 9

String odebraneDane = ""; //Pusty ciąg odebranych danych

void setup() {
  Serial.begin(9600); //Uruchomienie komunikacji
  pinMode(zielona, OUTPUT); //Konfiguracja wyjść
  pinMode(czerwona, OUTPUT);

  digitalWrite(zielona, LOW); //Wyłączamy diody
  digitalWrite(czerwona, LOW);
}

void loop() {
  if(Serial.available() > 0) { //Czy Arduino odebrano dane
    //Jeśli tak, to odczytujemy je do znaku końca linii i zapisz w zmiennej odebraneDane
    odebraneDane = Serial.readStringUntil('\n');

    if (odebraneDane == "zielona") { //Jeśli odebrano słowo "zielona"
      digitalWrite(zielona, HIGH); //To włączamy diodę zieloną
      delay(1000);
      digitalWrite(zielona, LOW);
    }

    if (odebraneDane == "czerwona") { //Jeśli odebrano słowo "czerwona"
      digitalWrite(czerwona, HIGH); //To włączamy diodę czerwoną
      delay(1000);
      digitalWrite(czerwona, LOW);
    }
  }
}
```

Źródła:

- 1) <http://elektronikab2b.pl/prezentacja-artykul/29465-komputery-jednoukladowe-som---robic-czy-kupic-gotowe#.WldPilxA6BE>
- 2) <https://www.mimuw.edu.pl/~marpe/arch/akmc.pdf> - Materiały do wykładu Mikrokomputery jednoukładowe, Marcin Peczarski
- 3) <https://pl.wikipedia.org/wiki/Arduino>
- 4) <http://forbot.pl/blog/artykuly/programowanie/kurs-arduino-2-podstawy-programowania-porty-io-id3648>
- 5) <https://proline.pl/?n=michigan-micro-mote-miniaturowy-komputer-wielkosci-ziarna-ryzu>
- 6) <http://esklep-drony.pl/pl/p/Arduino-Uno-Oryginal/624>
- 7) <http://www.zabezpieczenia.com.pl/sswin/system-sygnalizacji-w%C5%82amania-i-napadu-zaprojektowany-z-wykorzystaniem-mikrokontroler%C3%B3w-cz-2>
- 8) <http://www.arduino.org/learning/tutorials/ide-examples>

Linki aktywne dnia: 24.01.2017