

Nazwa implementacji: Funkcjonalność modułu-interfejsu - budowa sygnalizatora

Autor:

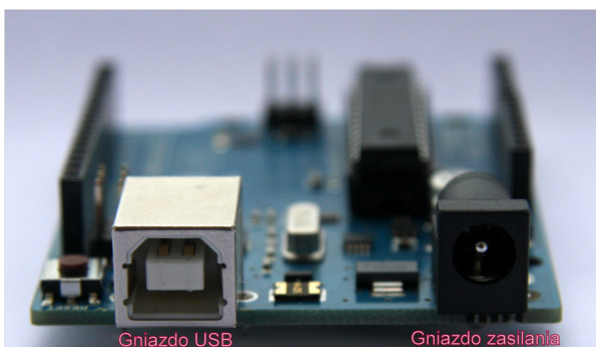
Krzysztof Bytow

Opis implementacji: Zastosowanie modułu-interfejsu oraz obsługa interaktywnego terminala Arduino IDE, służącego do programowania mikrokontrolera. Podłączenie i sterowanie diodami LED na przykładzie sygnalizacji świetlnej. Obsługa przycisków i sterowanie buzzerem. Zaimplementowanie kodu do sterowania diodą.

Wprowadzenie - opis wyprowadzeń, konfiguracja programu, podłączenie do komputera.



przewód usb wpinamy



w gniazdo USB →

Gniazdo USB

Gniazdo zasilania

Układ należy podłączyć przewodem USB do komputera.

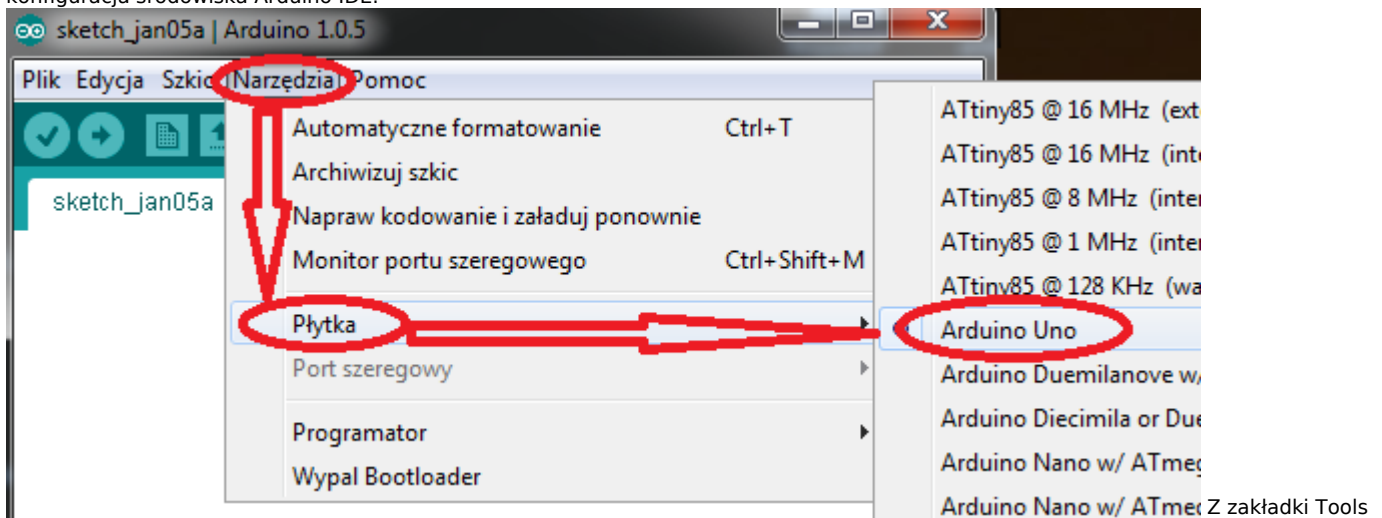


Opis wyprowadzeń Arduino UNO R3.

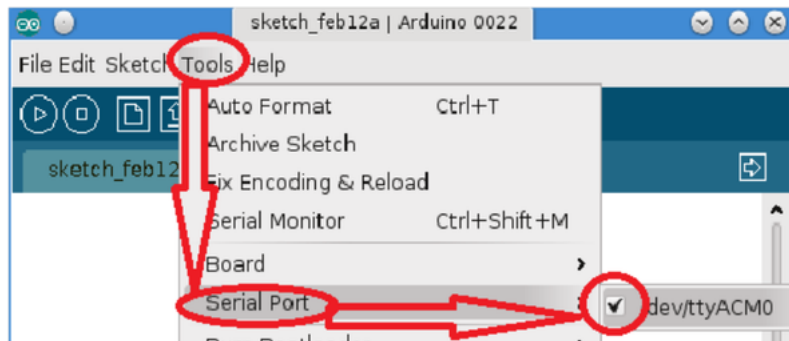


Uruchomienie i

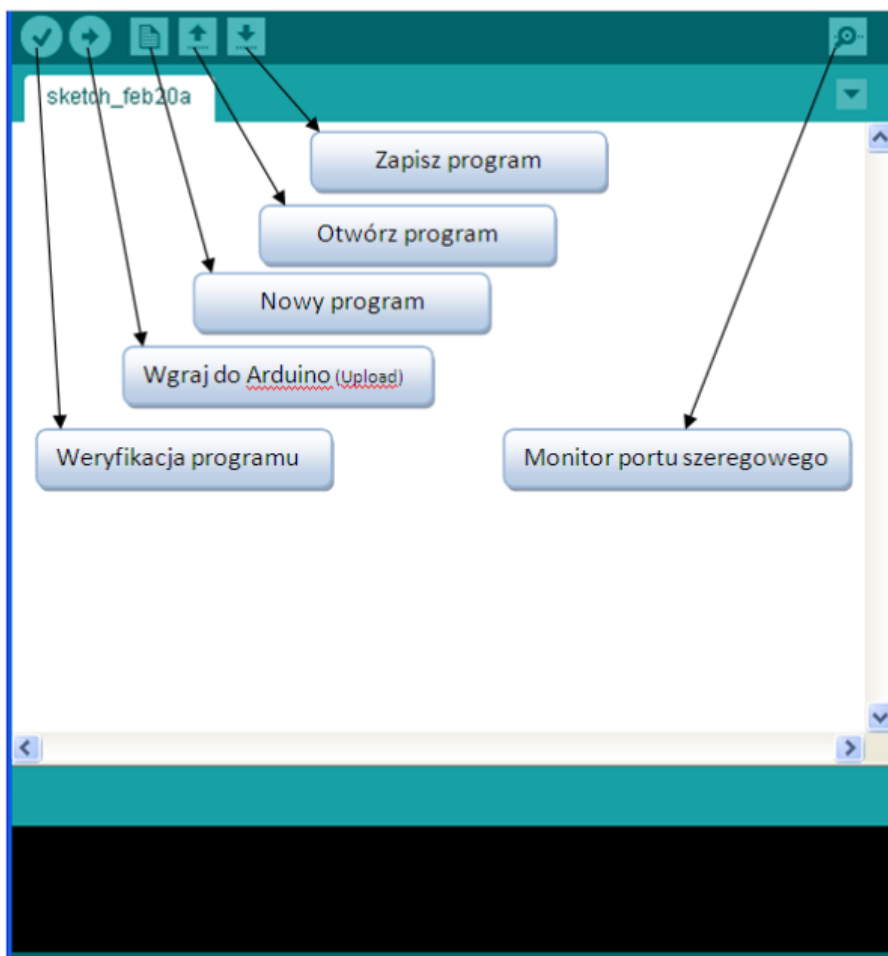
konfiguracja środowiska Arduino IDE.

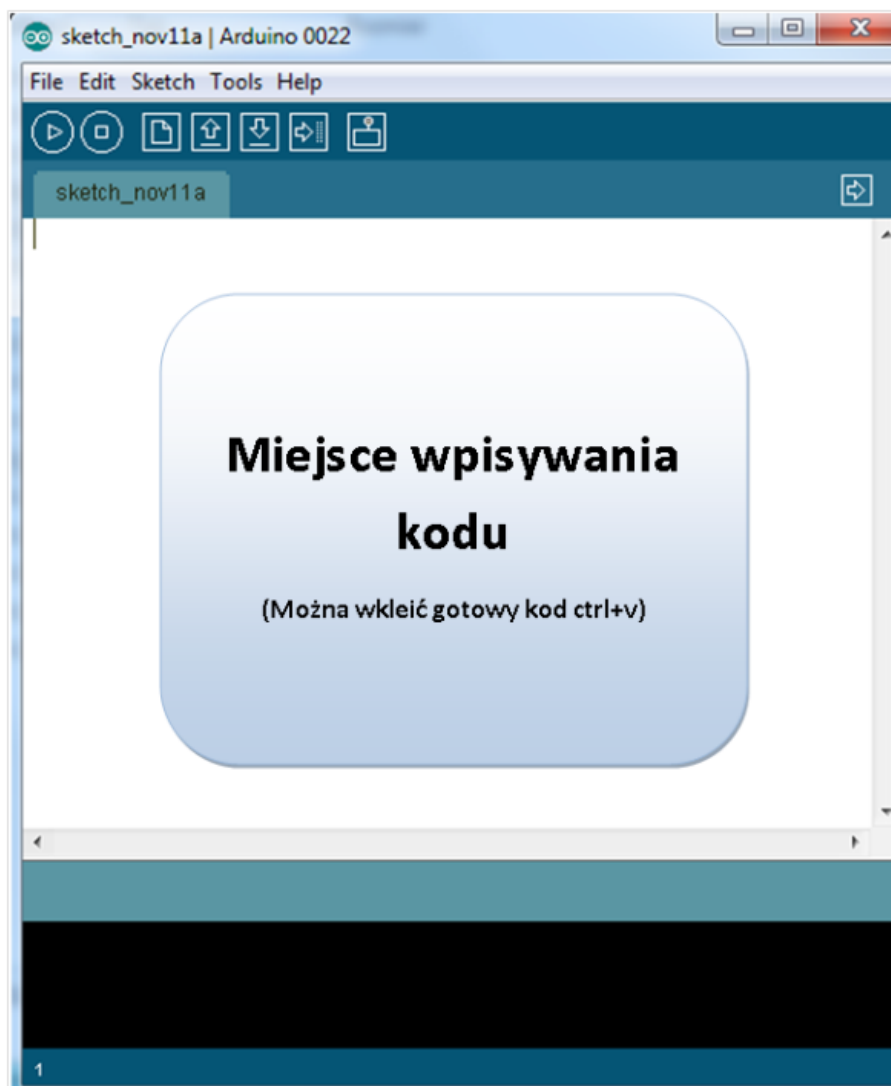


wyberamy Board, a następnie wersję, którą posiadamy (Arduino UNO).



Następnie należy wybrać port, pod który został podłączony układ (Tools → Serial port → odpowiedni port). Opis interfejsu środowiska Arduino IDE.





Schemat połączeń – sterowanie diodą led wbudowaną w układ oraz wyświetlenie tekstu na ekranie monitora



Kod implementacji:

```
int dioda = 13;           // przyznanie etykiety dla numeru pinu
int czas = 1000;         // tworzymy daną typu intiger
int x=10;

void setup()              // początkowa konfiguracja – część przygotowująca
{                          // układ do działania
  Serial.begin(9600);     // ustawienie prędkości komunikacji
  4
```

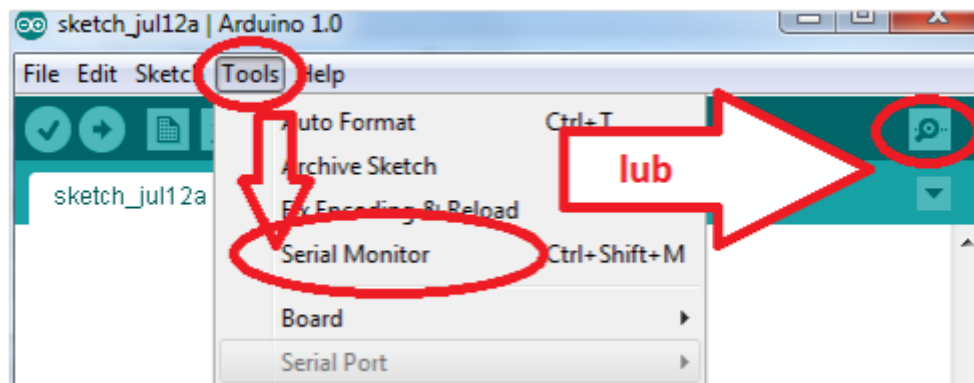
```

pinMode(dioda, OUTPUT);      // ustawienie pinu jako wyjście
}

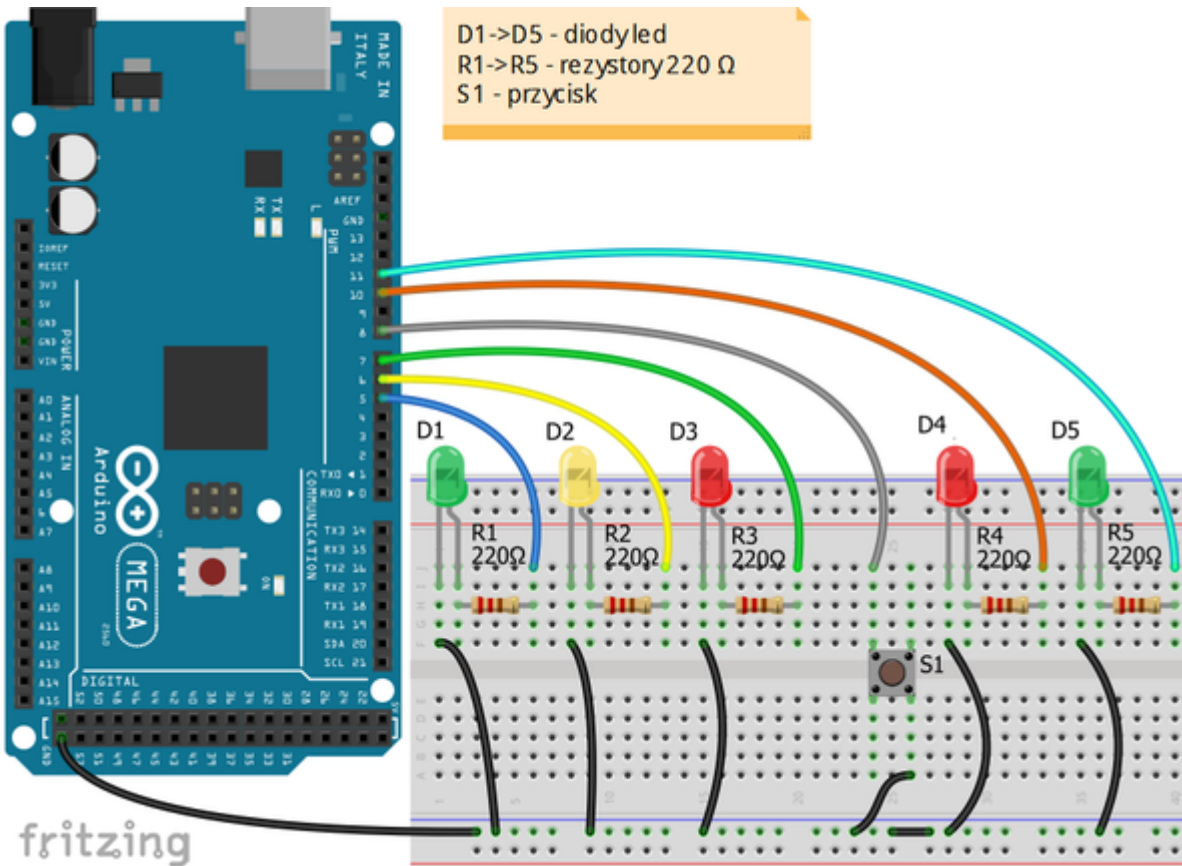
void loop()                  // główna pętla
{
Serial.println("Witaj - Arduino"); // wyświetlenie tekstu
delay(500);                  // czekaj 500ms = 0,5 s
Serial.print("Przypisana wartosc dla i = "); // wyświetl tekst
delay(500);                  // czekaj 500ms = 0,5 s
Serial.println(x);          // wyświetl wartość x
delay(500);                  // czekaj 500ms = 0,5 s
digitalWrite(dioda, HIGH);  // zapalenie diody - stan wysoki
Serial.println("Dioda swieci"); // wyświetlenie tekstu
delay(3000);                 // oczekiwanie 3000ms = 3 sekundy
digitalWrite(dioda, LOW);   // wyłączenie diody - stan niski
Serial.println("Dioda nieswieci"); // wyświetlenie tekstu
delay(1500);                 // oczekiwanie 1500ms = 1,5 sekundy
digitalWrite(dioda, HIGH);  // zapalenie diody - stan wysoki
Serial.println("Dioda swieci"); // wyświetlenie tekstu
delay(czas);                 // oczekiwanie 1000ms = 1 sekunda
digitalWrite(dioda, LOW);   // wyłączenie diody - stan niski
Serial.println("Dioda nieswieci"); // wyświetlenie tekstu
delay(500);                  // oczekiwanie 500ms = 0,5 sekundy
}

```

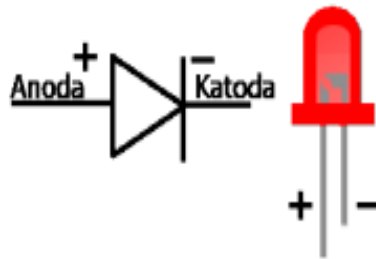
Po wgraniu kodu należy w programie Arduino IDE uruchomić Serial Monitor (lub terminal)



Komentarz (nie wpisywać do kodu implementacji): void setup() - początkowa konfiguracja - część przygotowująca; void loop() - główna pętla - wpisany program będzie wykonywany cyklicznie wciśnięcie przycisku reset powoduje uruchomienie kodu od początku; funkcja "pinMode" ustawia kierunek sygnału na pinie modułu (INPUT, OUTPUT); funkcja "digitalWrite" ustawia cyfrowy stan sygnału na pinie modułu (HIGH, LOW); funkcja "delay" czeka ustaloną w milisekundach długość czasu (1000 ms = 1 s); Schemat połączeń sterowanie diodami LED - na przykładzie sygnalizacji świetlnej



Uczeń/Uczennica po zestawieniu połączeń zgłasza nauczycielowi gotowość do sprawdzenia układu i wszystkich połączeń.

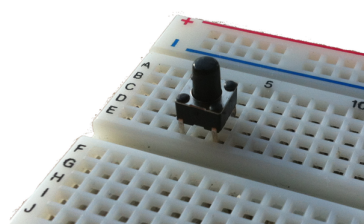


dioda LED czerwona

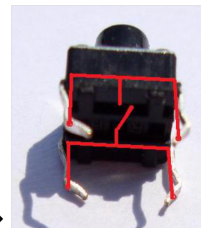


Rezystor 220 Ω

oznaczenie kodem barwnym rezystora 220 Ω

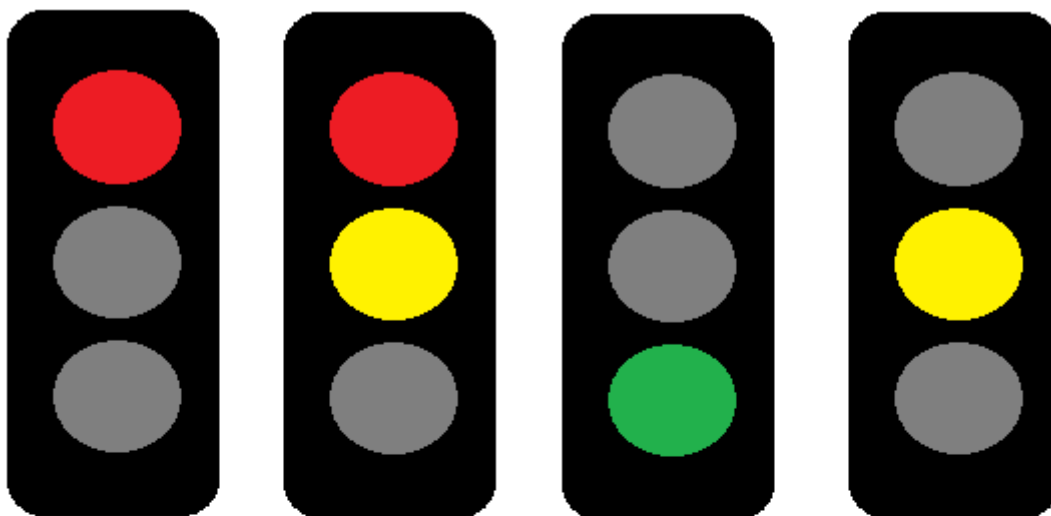


← button →

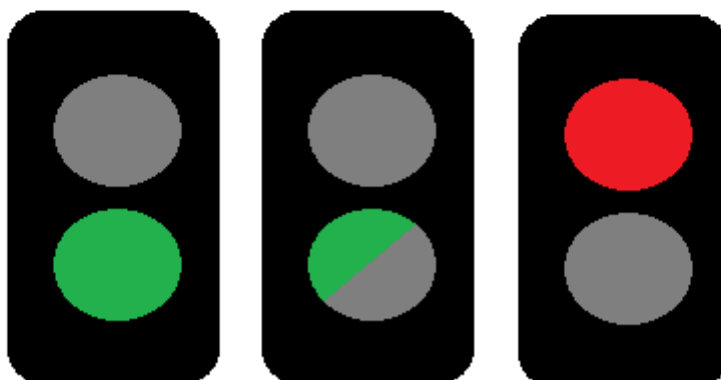




Sekwencje zmiany świateł:



sygnalizacja świetlna dla samochodów



sygnalizacja świetlna dla pieszych

Kod implementacji:

```

unsigned long czas;           //definiowanie zmiennych wykorzystanych w kodzie

void setup()
{
  for (int i=5;i<=12;i++) // w pętli ustawiamy piny od 5 do 12 jako wyjścia i ustawiamy stan niski
  {
    pinMode(i, OUTPUT);
    digitalWrite(i,LOW);
  }
  pinMode(8,INPUT);         // ustawiamy Pin 8 jako wejście
  digitalWrite(8,HIGH);     // pullup
  digitalWrite(10,HIGH);    // zapalamy diody wpięte do Pinu 10 i 5
  digitalWrite(5,HIGH);
}
7

```





```
void loop()
{
  if (digitalRead(8)==LOW && (millis() - czas) > 5000)      // funkcja warunkowa
  {
    przejscie();      // gdy zostanie wciśnięty button wywołanie funkcji przejście
  }
}

void przejscie()
{
  delay(800);      // czekaj wyznaczony czas
  off();      // wywołanie funkcji gaszącej diody
  digitalWrite(6,HIGH);
  delay(2000);      // czekaj 2 sekundy
  off();      // wywołanie funkcji gaszącej diody
  digitalWrite(7,HIGH);
  delay(2000);      // czekaj wyznaczony czas
  digitalWrite(10,LOW);
  digitalWrite(11,HIGH);
  delay(5000);      // czekaj wyznaczony czas
  for (int x=0;x<10;x++)      // mruganie diody zielonej dla pieszych
  {
    digitalWrite(11,HIGH);
    delay(250);      // czekaj wyznaczony czas
    digitalWrite(11,LOW);
    delay(250);      // czekaj wyznaczony czas
  }
  digitalWrite(10,HIGH);
  delay(3000);      // czekaj wyznaczony czas
  digitalWrite(6,HIGH);
  delay(2000);      // czekaj wyznaczony czas
  off();      // wywołanie funkcji gaszącej diody
  digitalWrite(5,HIGH);
  czas=millis();      // millis() zwraca liczbę milisekund od czasu jak Arduino wykonuje
}      // bieżący program

void off()      // gasimy diody od pinu 5 do 7 ustawiając stan niski
{
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
}
```

Filmy instruktażowe: <http://youtu.be/3nuzFeVfIUE> <http://youtu.be/EUIIq-zlhnw>

