

```
digitalWrite(R,  
digitalWrite(GG,  
digitalWrite(B,  
t_ruch = millis(  
while((millis()
```



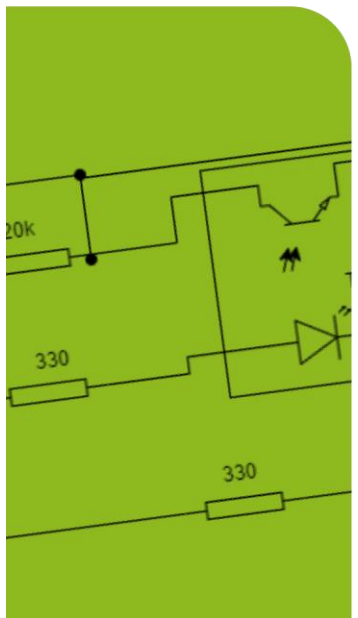
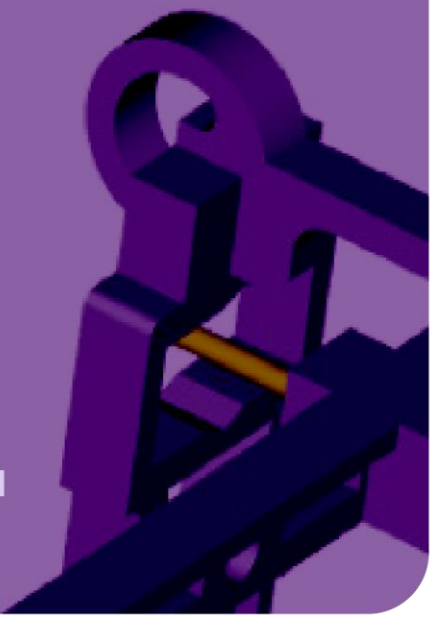
Wojciech Kolarz

PROJEKT

POTENCJOMETR LED SERWO

DIY

MODELOWANIE 3D
PROGRAMOWANIE
PODSTAWY ELEKTRONIKI



Informacje uzupełniające do materiału filmowego.

Elementy i materiały użyte do konstrukcji urządzenia:

- Elementy drukowane: filament PLA, standardowe ustawienia wydruku, wypełnienie dowolne (zastosowane w zaprezentowanym urządzeniu: 10%).
- Diody LED – standardowe czerwone i niebieskie 5mm.
- Rezystory:
 - 470 Ω - 3 szt.
 - 100 Ω - 2 szt.
- Kondensatory: 100nF – 3 szt.
- Potencjometry:
 - 10 k Ω liniowy, obrotowy, średnica zewnętrzna 17 mm – 1 szt.
 - 10 k Ω logarytmiczny, obrotowy, średnica zewnętrzna 17 mm – 2 szt.
- Arduino UNO R3.
- Serwomechanizm modelarski SG-90.
- Wkręty – dostępne w komplecie z serwem, wkręty (2-3 mm) – np. wkręty do drewna.
- Przewody, drut miedziany, klej na gorąco, koszulki termokurczliwe.

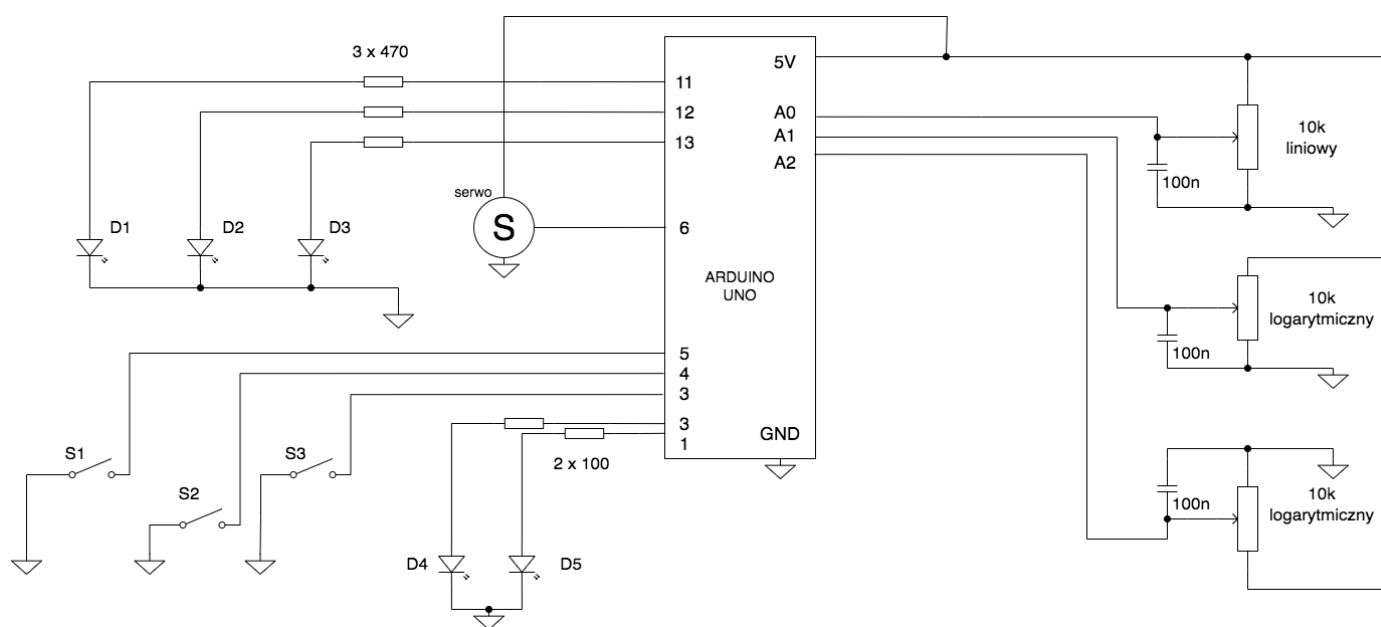
Schemat urządzenia.

Elementy można podłączyć do innych, niż przedstawione w opisie, pinów Arduino.

Trzeba pamiętać tylko o tym, aby diody D4 i D5 podłączone były do pinów obsługujących PWM (piny te oznaczone są znakiem \sim).

Serwomechanizm może być podłączony do dowolnego pniu. **UWAGA: użycie biblioteki serwa wyłącza obsługę PWM na pinach 9 i 10.**

Rezystory dobrane są do rodzaju typowych diod LED – D1, D2, D3 – czerwone D3, D4 – niebieskie. W przypadku zastosowania diod LED o innych kolorach, należy dobrać inne wartości rezystancji zastosowanych rezystorów.

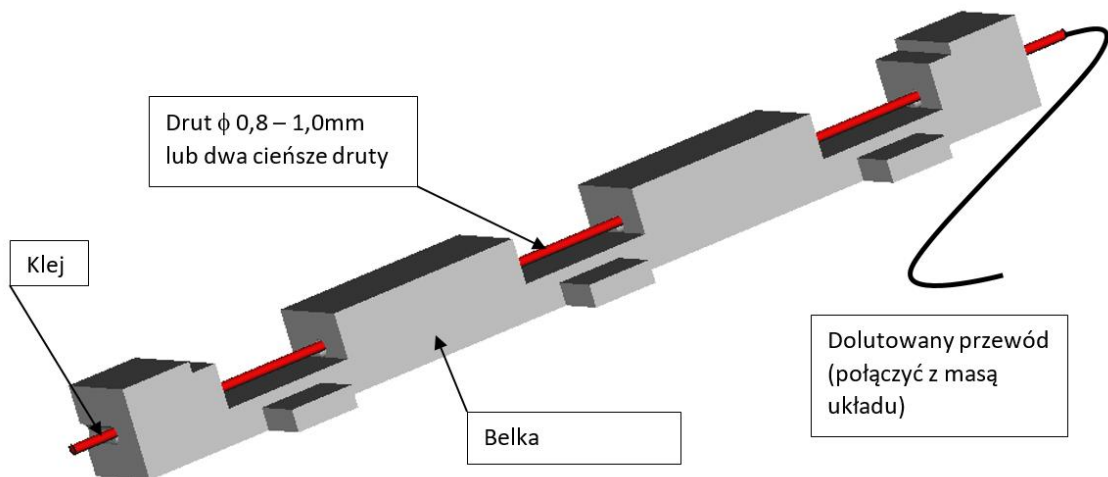


Montaż urządzenia.

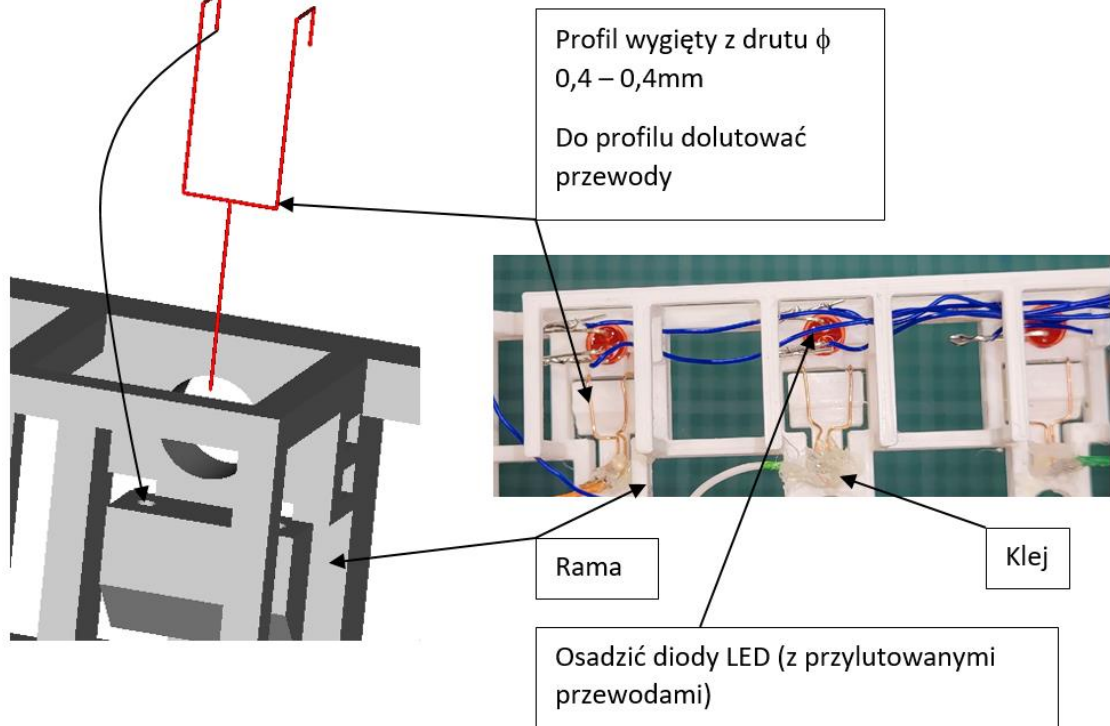
Załączone pliki .stl umożliwiają wydrukowanie podstawowych elementów urządzenia. Można samodzielnie zaprojektować własną postać niezbędnych elementów.

W urządzeniu zostały użyte typowe, dostępne na rynku pokręta potencjometrów. Pokręta można również wydrukować (w internecie dostępne są darmowe pliki .stl). Korzystając z załączonych plików, montaż należy wykonać w kolejności przedstawionej poniżej. Wskaźnik serwa został wykonany z kolorowej tektury, przyklejonej do orczyka serwa. Na pokrywę górną nałożona została podziałka (wydrukowana na zwykłym papierze + warstwa cienkiej folii – te elementy można wykonać indywidualnie wg. własnego pomysłu).

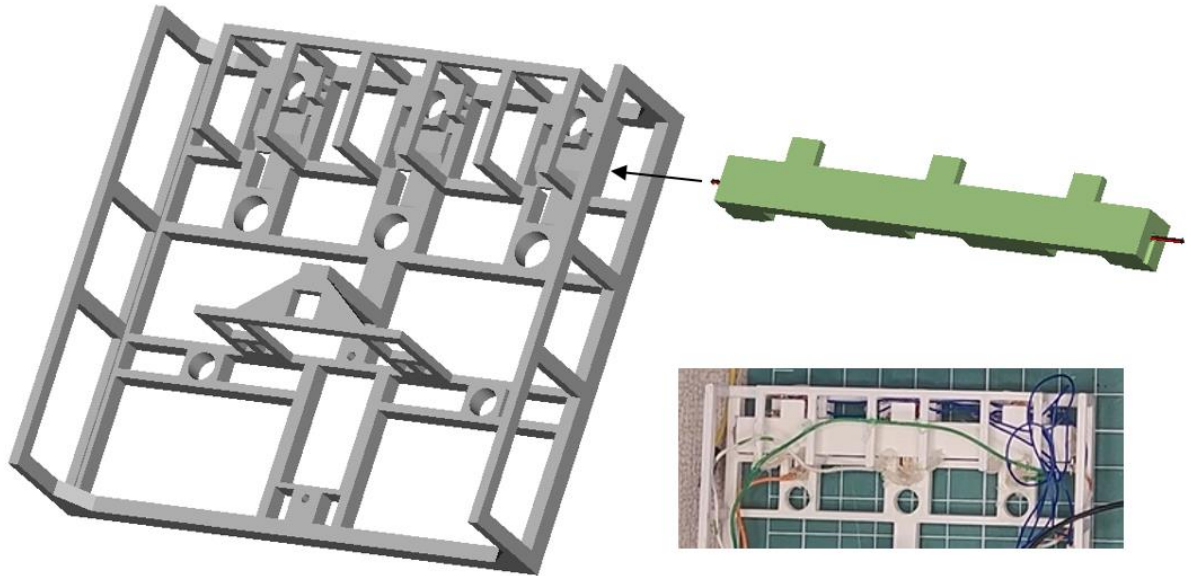
1



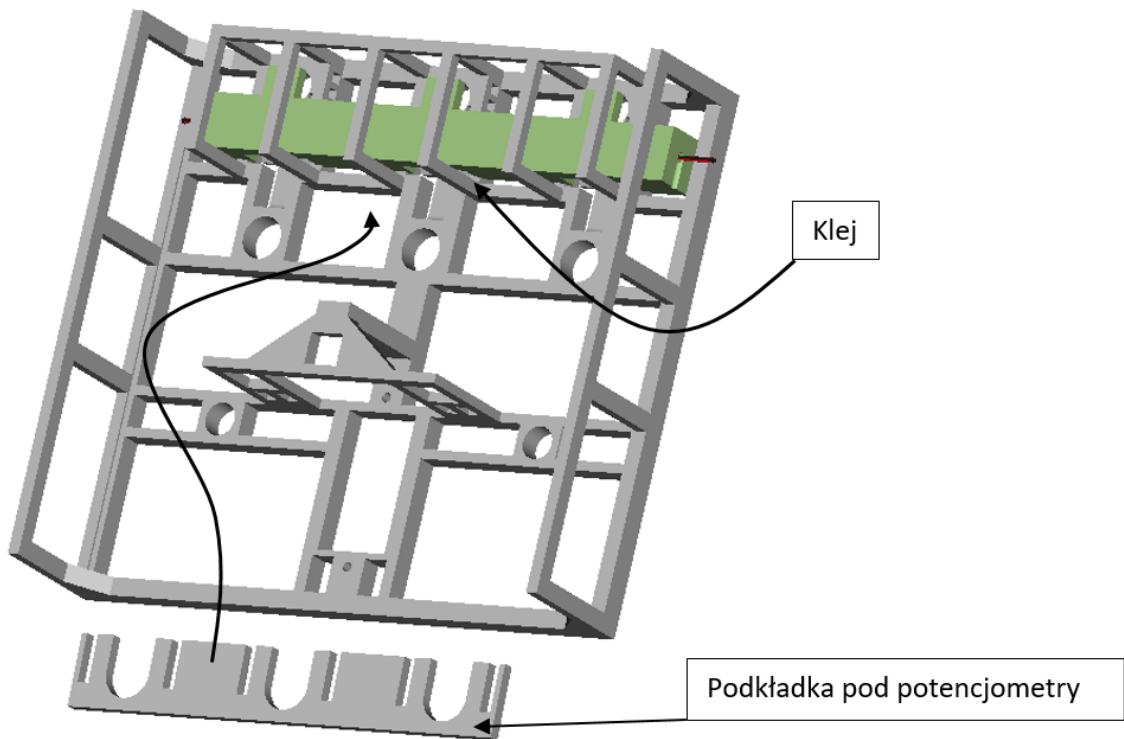
2



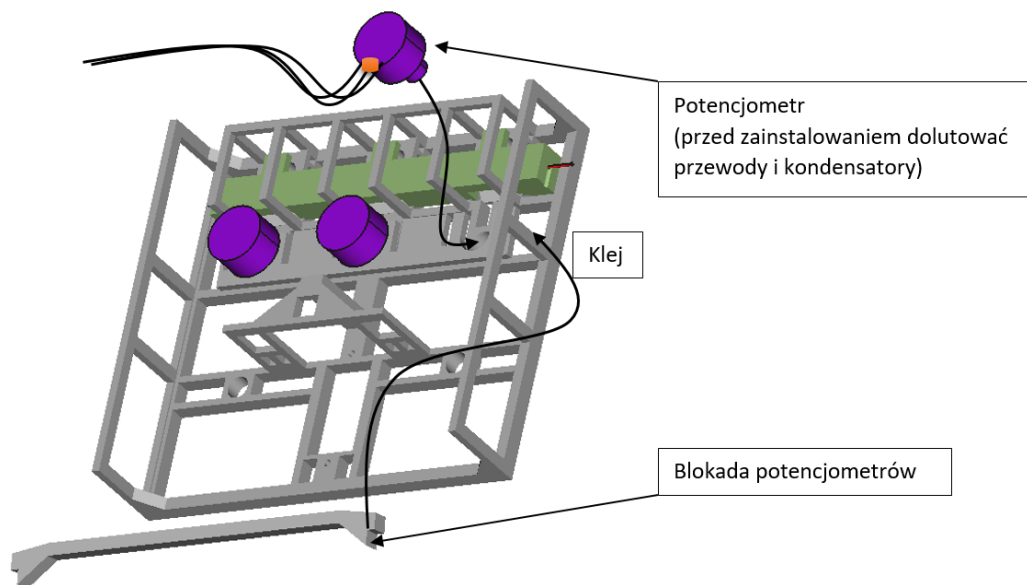
3



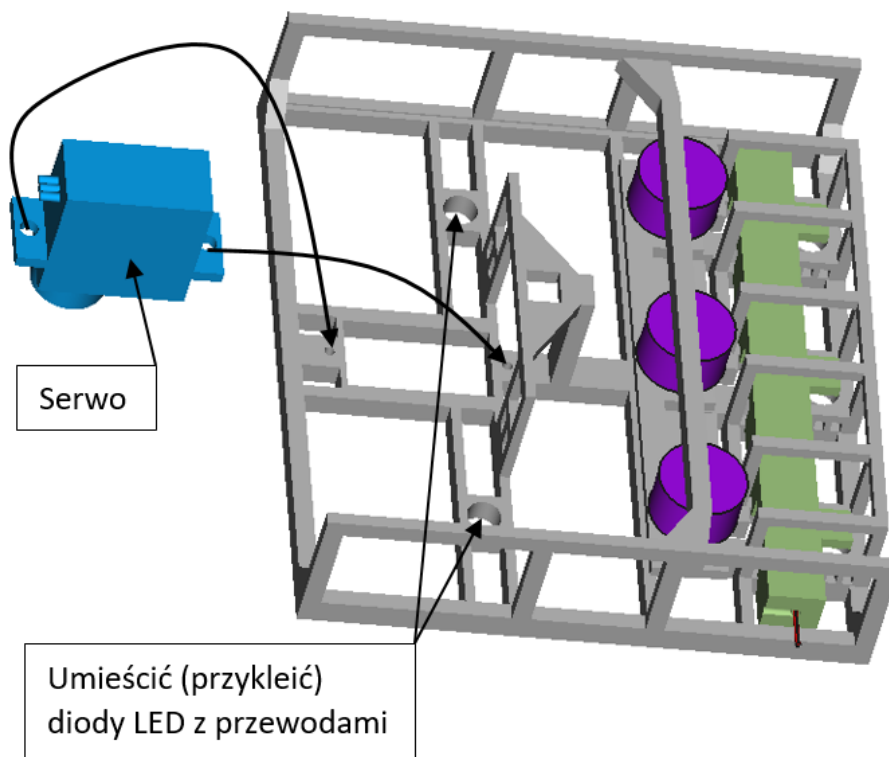
4

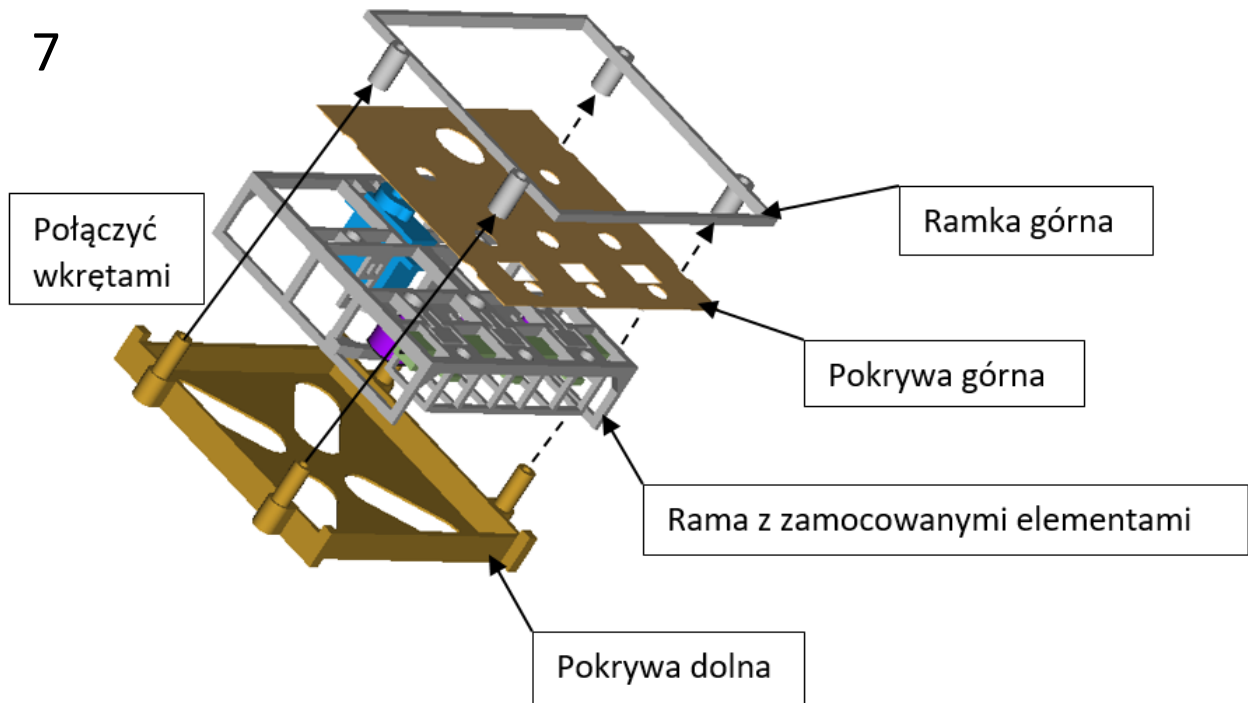


5



6





Uwagi do montażu.

1. Kondensatory filtrujące można dolutować bezpośrednio przy potencjometrach, jednak lepiej będą spełniały swoją funkcję, jak przylutowane będą jak najbliżej pinów Arduino.
2. Wszystkie miejsca lutowane należy zaizolować np. stosując koszulki termokurczliwe lub w dowolny inny sposób.
3. Po wydruku może być konieczne delikatne przeszlifowanie pilnikiem niektórych elementów – przed ostatecznym montażem (sklejeniem) sprawdź pasowanie elementów.
4. Sprawdź rzeczywisty kąt obrotu serwa (czy ma on pełne 180°), ewentualnie uwzględnij to w podziałce umieszczonej na pokrywie górnej (załączony plik do wydrukowania podziałki został przygotowany przy założeniu, że serwo posiada możliwość obrotu o 180°).

Pogram

```

1. #include <Servo.h>
2. // definicje nazw pinów włączników
3. #define S1 2
4. #define S2 3
5. #define S3 4
6. // definicje nazw pinów diod sygnalizujących
   stan pracy
7. #define D1 11
8. #define D2 12
9. #define D3 13
10. // definicje nazw pinów diod o regulowanym
    natężeniu światła
11. #define D4 5
12. #define D5 6
13. // definicje nazw pinów potencjometrów
14. #define P1 A0
15. #define P2 A1
16. #define P3 A2
17. int sekcja = 0; // numer wybranej sekcji
    (potencjometr logarytmiczny/linowy)
18. int pwm = 0; //aktualna wartość PWM
19. int p = 0; // wartość odczytana z
    potencjometru
20. int pozycja; // pozycja ustawienia serwa
21. Servo serwomechanizm;

45. void loop() {
46. if (not(digitalRead(S1))) sekcja = 1;
47. if (not(digitalRead(S2))) sekcja = 2;
48. if (not(digitalRead(S3))) sekcja = 3;
49. switch (sekcja){
50. case 1:
51. p = analogRead(P1);
52. digitalWrite(D1,HIGH);
53. digitalWrite(D2, LOW);
54. digitalWrite(D3, LOW);
55. break;
56. case 2:
57. p = analogRead(P2);
58. digitalWrite(D1,LOW);
59. digitalWrite(D2, HIGH);
60. digitalWrite(D3, LOW);

22. void setup() {
23. serwomechanizm.attach(1);
24. pinMode(S1, INPUT_PULLUP);
25. pinMode(S2, INPUT_PULLUP);
26. pinMode(S3, INPUT_PULLUP);
27. pinMode(D1, OUTPUT);
28. pinMode(D2, OUTPUT);
29. pinMode(D3, OUTPUT);
30. pinMode(D4, OUTPUT);
31. pinMode(D5, OUTPUT);
32. //sygnalizacja uruchomienia urządzenia
33. digitalWrite(D1,HIGH);
34. delay(500);
35. digitalWrite(D1,LOW);
36. digitalWrite(D2,HIGH);
37. delay(500);
38. digitalWrite(D2,LOW);
39. digitalWrite(D3,HIGH);
40. delay(500);
41. digitalWrite(D3,LOW);
42. pozycja = 0;
43. serwomechanizm.write(pozycja);
44. }

61. break;
62. case 3:
63. p = analogRead(P3);
64. digitalWrite(D1,LOW);
65. digitalWrite(D2, LOW);
66. digitalWrite(D3, HIGH);
67. break;
68. }
69. if (p>1010) p = 1010; // można użyć funkcji
    constrain()
70. pwm = map(p,0,1010,0,255);
71. analogWrite(D4,pwm);
72. analogWrite(D5,pwm);
73. pozycja=map(p,0,1010,0,180);
74. serwomechanizm.write(pozycja);
75. delay(100);
}

```

Zaprezentowany program jest programem prostym. Wzbogacając program o dodatkowe treści, a tym samym wzbogacając funkcjonalności urządzenia można wykorzystywać urządzenie do innych prób i doświadczeń związanych z programowaniem Arduino.