

PROJEKT

STEROWNIKI ROBOTÓW

---

## Raport

Humanistycznie upośledzony robot  
akrobatyczny

HURA

---

*Skład grupy:*

Albert LIS, 235534

Michał MORUŃ, 235986

*Termin:* sr TP15

*Prowadzący:*

mgr inż. Wojciech DOMSKI

10 czerwca 2019

# Spis treści

<b>1</b>	<b>Konfiguracja mikrokontrolera</b>	<b>2</b>
1.1	Konfiguracja pinów . . . . .	2
<b>2</b>	<b>Urządzenia zewnętrzne</b>	<b>2</b>
<b>3</b>	<b>Projekt elektroniki</b>	<b>2</b>
3.1	Schemat elektryczny połączenia z fotorezystorami . . . . .	2
3.2	Regulacja położenia wertykalnego . . . . .	2
3.3	Regulacja położenia horyzontalnego . . . . .	3
<b>4</b>	<b>Konstrukcja mechaniczna</b>	<b>4</b>
<b>5</b>	<b>Opis działania programu</b>	<b>5</b>
5.1	Schemat działania programu sterującego platformą . . . . .	5
5.2	Funkcja czytająca natężenie światła . . . . .	5
5.3	Funkcja sterująca silnikiem krokowym . . . . .	5
5.4	Funkcja sterująca serwomechanizmem platformy . . . . .	6
5.5	Funkcje odpowiadające za sprawdzenie czy różnica odczytów jest większa od tolerancji . .	6
<b>6</b>	<b>Zrealizowane zadania</b>	<b>6</b>
6.1	Wizualizacja . . . . .	6
<b>7</b>	<b>Urządzenia zewnętrzne</b>	<b>7</b>
<b>8</b>	<b>Podsumowanie</b>	<b>7</b>
	<b>Bibilografia</b>	<b>8</b>

# 1 Konfiguracja mikrokontrolera

## 1.1 Konfiguracja pinów

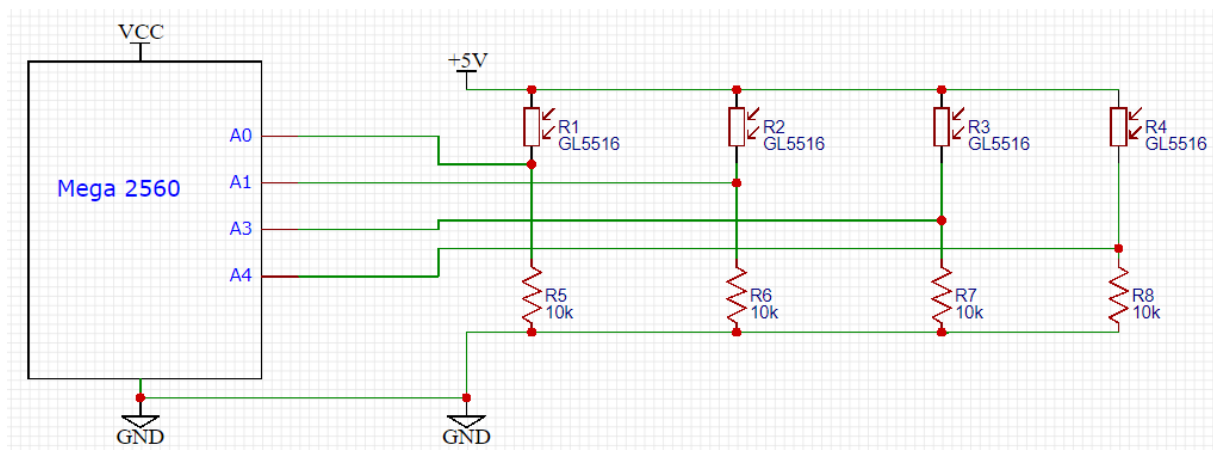
PIN	Tryb pracy	Funkcja/etykieta
A0	Analog Input	LEFT_DOWN_PIN
A1	Analog Input	RIGHT_DOWN_PIN
A2	Analog Input	LEFT_UP_PIN
A3	Analog Input	RIGHT_UP_PIN
0	Rx	RX
1	Tx	TX
5	Digital Output	SERVO_PIN
6	Digital Output	DIRECTION
7	Digital Output	STP

Tabela 1: Konfiguracja pinów mikrokontrolera

## 2 Urządzenia zewnętrzne

## 3 Projekt elektroniki

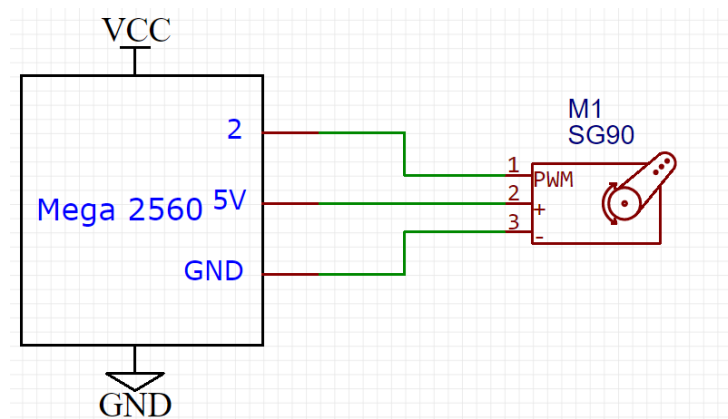
### 3.1 Schemat elektryczny połączenia z fotorezystorami



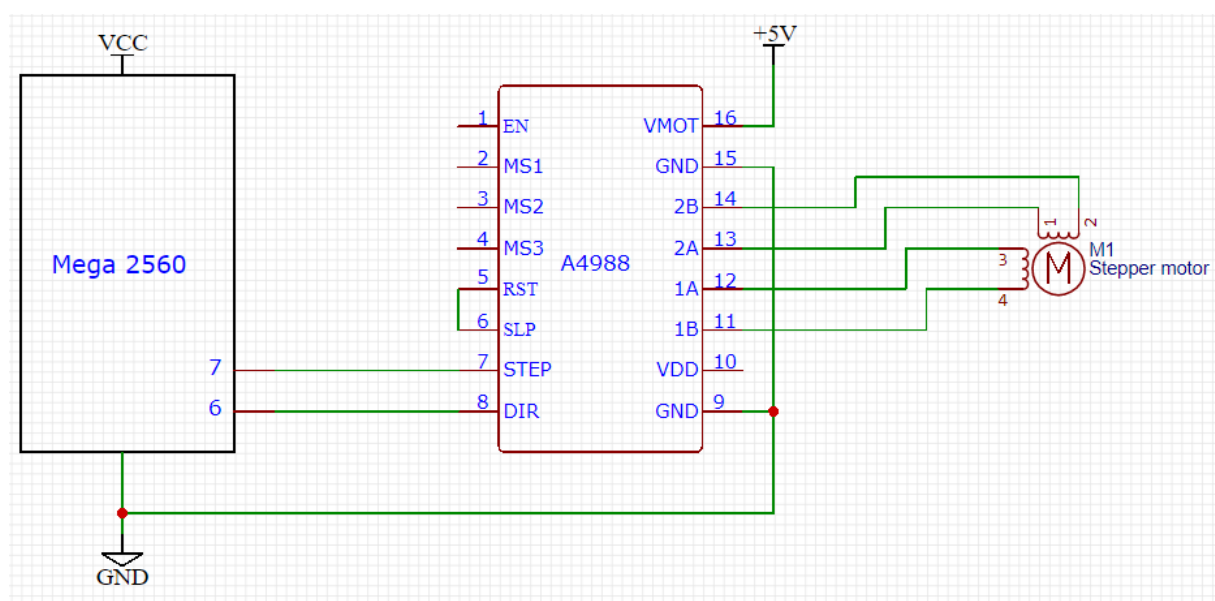
Rysunek 1: Schemat elektryczny połączenia z fotorezystorami

### 3.2 Regulacja położenia wertykalnego i horyzontalnego

Do regulacji położenia wertykalnego robota wykorzystano serwo zamontowane na platformie. Jest ono sterowane za pomocą sygnału PWM z Arduino. Obok silnika krokowego zapewnia poruszanie się robota w kierunku światła. Serwo odpowiada za dokładne ustawienie prostopadle do słońca, podwyższając lub obniżając ścianki na których są zamontowane fotorezystory, natomiast silnik krokowy obraca postawę platformy w kierunku światła.

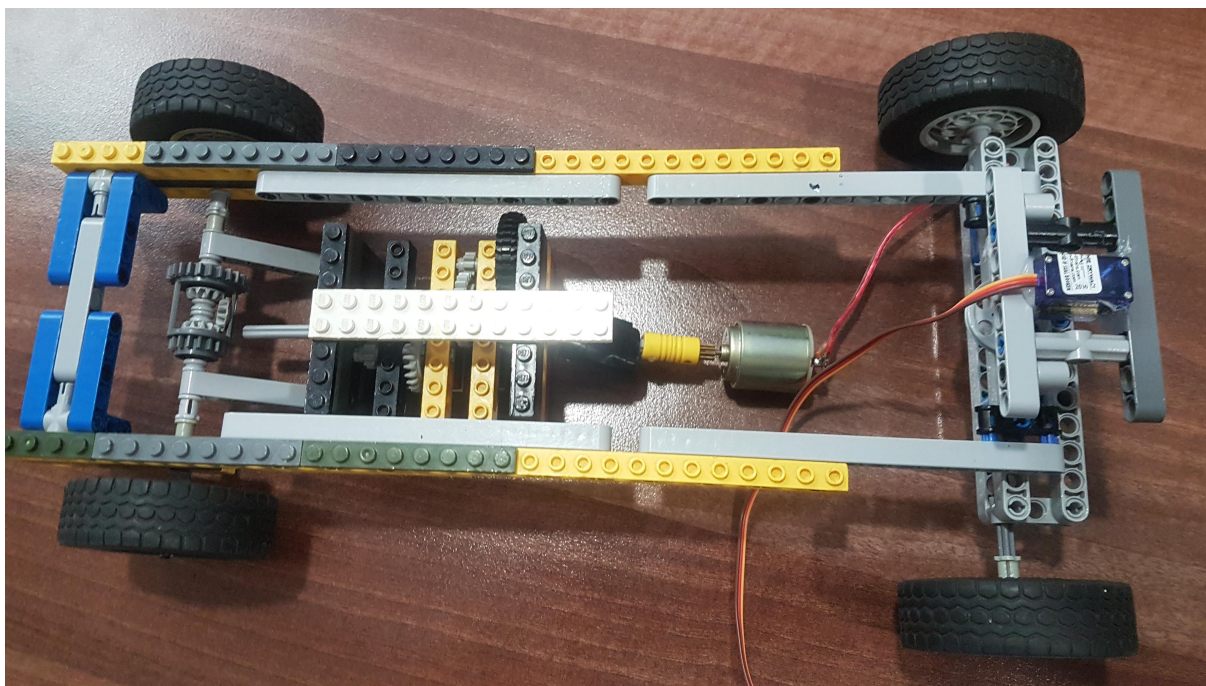


Rysunek 2: Schemat elektryczny sterowania serwomechanizmem

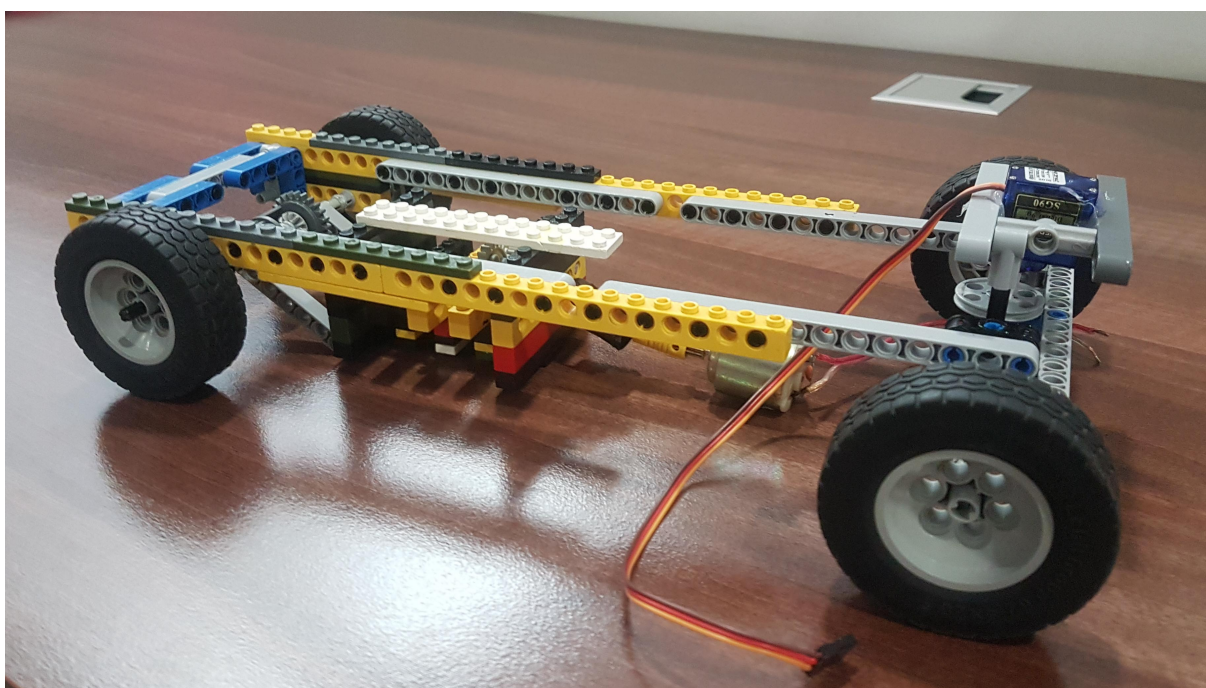


Rysunek 3: Schemat elektryczny sterowania silnikiem krokowym

## 4 Konstrukcja mechaniczna



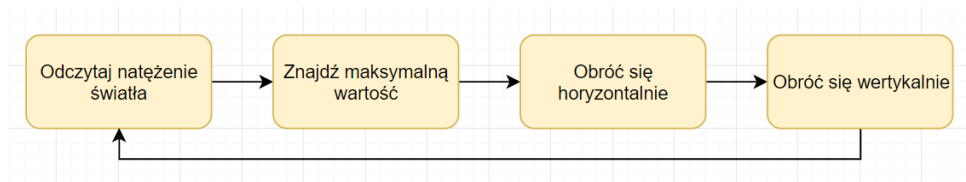
Rysunek 4: Zdjęcie części mechanicznej nr 1



Rysunek 5: Zdjęcie części mechanicznej nr 2

## 5 Opis działania programu

### 5.1 Schemat działania programu sterującego platformą



Rysunek 6: Schemat działania programu

### 5.2 Funkcja czytająca natężenie światła

Jest odpowiedzialna za odczyt wartości i umieszczenie ich w tablicy.

```
1 void ReadLight()
2 {
3     values[0] = analogRead(LEFT_DOWN_PIN);
4     values[1] = analogRead(RIGHT_DOWN_PIN);
5     values[2] = analogRead(LEFT_UP_PIN);
6     values[3] = analogRead(RIGHT_UP_PIN);
7 }
```

### 5.3 Funkcja sterująca silnikiem krokowym

Odpowiada za ruch platformy lewo-prawo. Realizuje poruszanie się w kierunku najintensywniejszego odczytu natężenia światła.

```
1 void SetStepperPosition()
2 {
3     //Jeśli różnica przekracza tolerancję
4     if (HorizontalDiff()) {
5         //Jeśli platforma jest obrócona wertykalnie w drugą stronę zmienia kierunek
6         //lewo/prawo
7         if (sposition > 90) {
8             //jeśli maksymalny odczyt z lewej strony
9             if ((pos == 0) || (pos == 2)) {
10                 digitalWrite(DIRECTION, HIGH);
11                 ++StepCounter;
12             }
13             else {
14                 digitalWrite(DIRECTION, LOW);
15                 --StepCounter;
16             }
17         }
18         else {
19             if ((pos == 1) || (pos == 3)) {
20                 digitalWrite(DIRECTION, HIGH);
21                 ++StepCounter;
22             }
23             else {
24                 digitalWrite(DIRECTION, LOW);
25                 --StepCounter;
26             }
27         }
28         //Wykonaj krok
29         digitalWrite(STP, state);
30         state = !state;
31     }
32 }
```

## 5.4 Funkcja sterująca serwomechanizmem platformy

```
1 void SetServoPosition()
2 {
3     //Jeśli przekracza tolerancję
4     if (VerticalDiff()) {
5         //Jeśli maksymalny odczyt na dole
6         if ((pos == 0) || (pos == 1)) {
7             if (sposition > 0)
8                 serwo.write(--sposition);
9         }
10        else {
11            if (sposition < 180)
12                serwo.write(++sposition);
13        }
14    }
15 }
```

## 5.5 Funkcje odpowiadające za sprawdzenie czy różnica odczytów jest większa od tolerancji

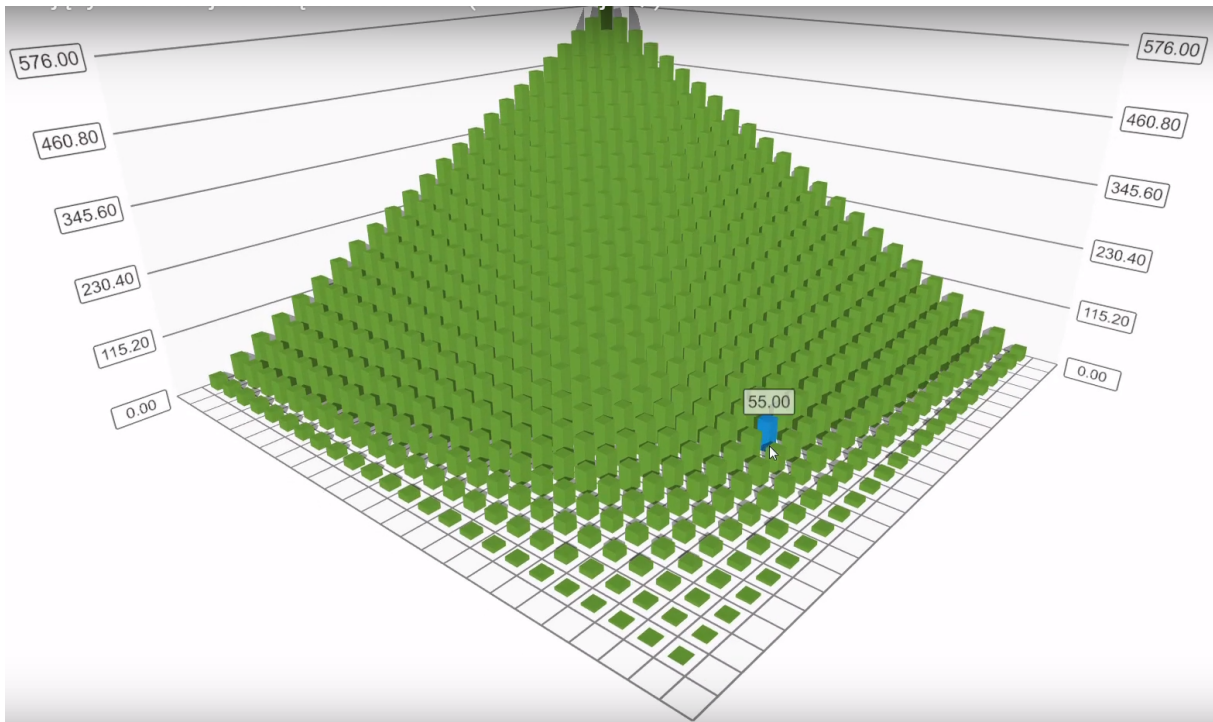
```
1 inline bool VerticalDiff()
2 {
3     upMax = ( values[2] > values[3] ? values[2] : values[3] );
4     downMax = ( values[0] > values[1] ? values[0] : values[1] );
5     return ( abs(upMax - downMax) > TOLERANCE ? true : false );
6 }
7
8 inline bool HorizontalDiff()
9 {
10    leftMax = ( values[0] > values[2] ? values[0] : values[2] );
11    rightMax = ( values[1] > values[3] ? values[1] : values[3] );
12    return ( abs(leftMax - rightMax) > TOLERANCE ? true : false );
13 }
```

# 6 Zrealizowane zadania

## 6.1 Wizualizacja

Udało się zrealizować wizualizację ukazującą natężenie światła w określonym miejscu w przestrzeni zrealizowaną dla przykładowych danych. Wizualizacja zakłada, że robot będzie poruszał się po polu prostokąta raz przy razie zbierając w ramach możliwości jak najdokładniejsze dane.

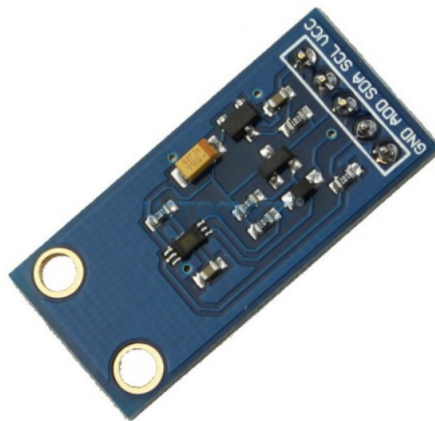




Rysunek 7: Przykładowa wizualizacja

## 7 Urządzenia zewnętrzne

Wykorzystywanym w projekcie urządzeniem zewnętrznym jest czujnik natężenia światła – GY-30-BH1750. Czujnik został zamontowany na górze platformy z fotorezystorami w celu zbierania informacji potrzebnych do wizualizacji.



Rysunek 8: Czujnik wykorzystany na platformie

## 8 Podsumowanie

Udało się zrealizować większość zadań. Nastąpiły drobne zmiany koncepcyjne jak użycie potencjometru do regulacji prędkości obrotowej napędu. To będzie wymagać mniejszej ingerencji gdy będziemy projektować regulator PID.



## Literatura

- [1] Krzysztof Amborski, Andrzej Murusak: *Teoria sterowania w ćwiczeniach*, ('78)
- [2] Jerzy Brzózka: *Regulatory i układy automatyki*, (2004)
- [3] Krzysztof Tchoń: *Manipulatory i roboty mobilne : modele, planowanie ruchu, sterowanie*, (2000)