

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

27 maja 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 . . . . .	2
3.2	Regulacja prędkości napędu . . . . .	2
<b>4</b>	<b>Konstrukcja mechaniczna</b>	<b>3</b>
<b>5</b>	<b>Opis działania programu</b>	<b>4</b>
5.1	Schemat działania programu . . . . .	4
5.2	Funkcja czytająca natężenie światła . . . . .	4
5.3	Funkcja sterująca silnikiem krokowym . . . . .	4
5.4	Funkcja sterująca serwomechanizmem platformy . . . . .	5
5.5	Funkcje odpowiadające za sprawdzenie czy różnica odczytów jest większa od tolerancji . . . . .	5
<b>6</b>	<b>Zadania niezrealizowane</b>	<b>6</b>
<b>7</b>	<b>Podsumowanie</b>	<b>6</b>
	<b>Bibilografia</b>	<b>7</b>

# 1 Konfiguracja mikrokontrolera

## 1.1 Konfiguracja pinów

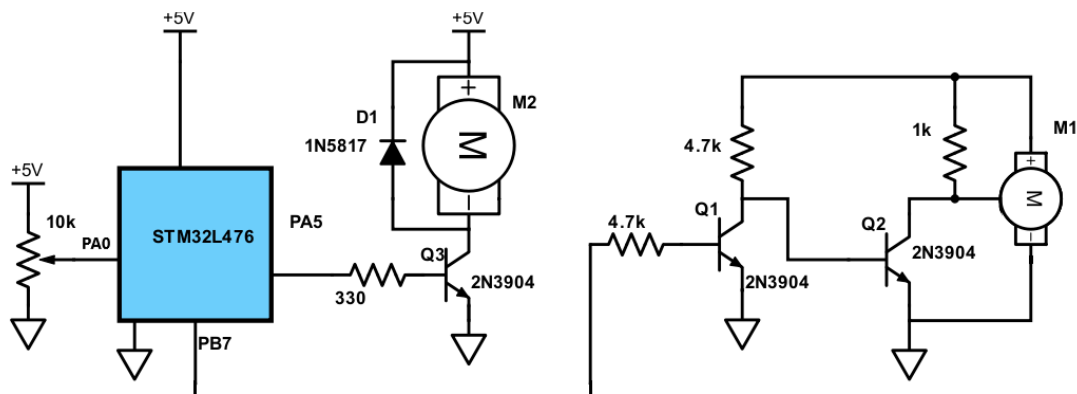
PIN	Tryb pracy	Funkcja/etykieta
A0	Analog Input	LEFT_DOWN_PIN
A1	Analog Input	RIGHT_DOWN_PIN
A2	Analog Input	LEFT_UP_PIN
A3	Analog Input	RIGHT_UP_PIN
0	Rx	RX
1	Tx	TX
5	Digital Output	SERVO_PIN
6	Digital Output	DIRECTION
7	Digital Output	STP

Tabela 1: Konfiguracja pinów mikrokontrolera

## 2 Urządzenia zewnętrzne

## 3 Projekt elektroniki

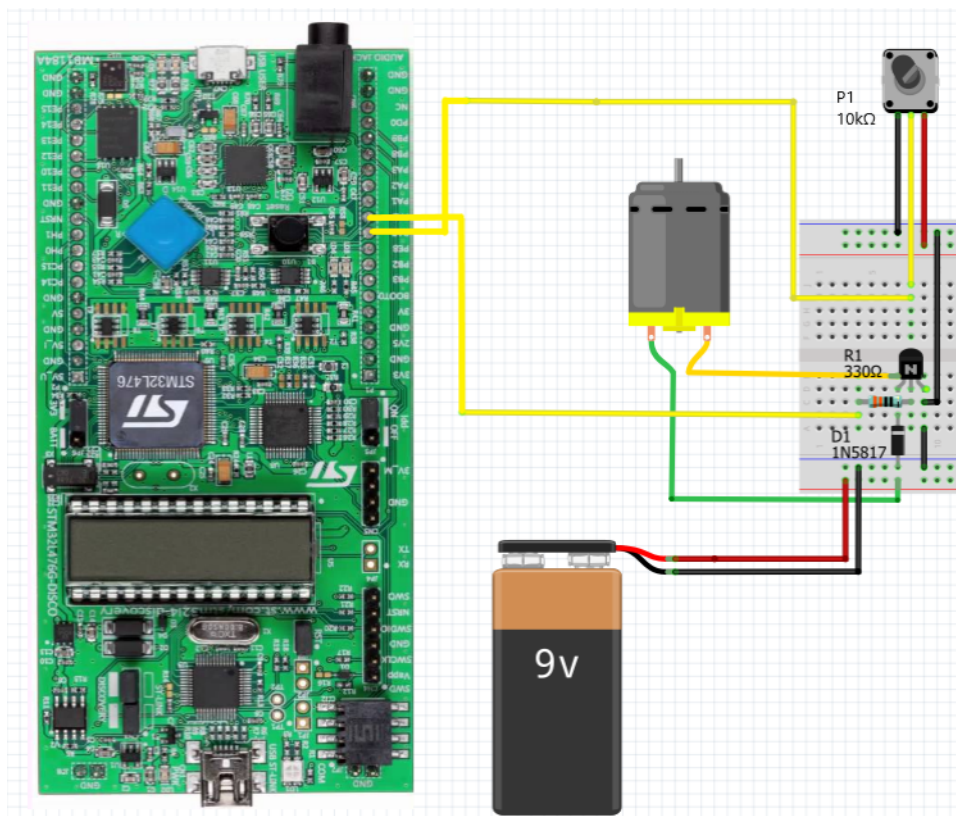
### 3.1 Schemat elektryczny



Rysunek 1: Schemat elektryczny

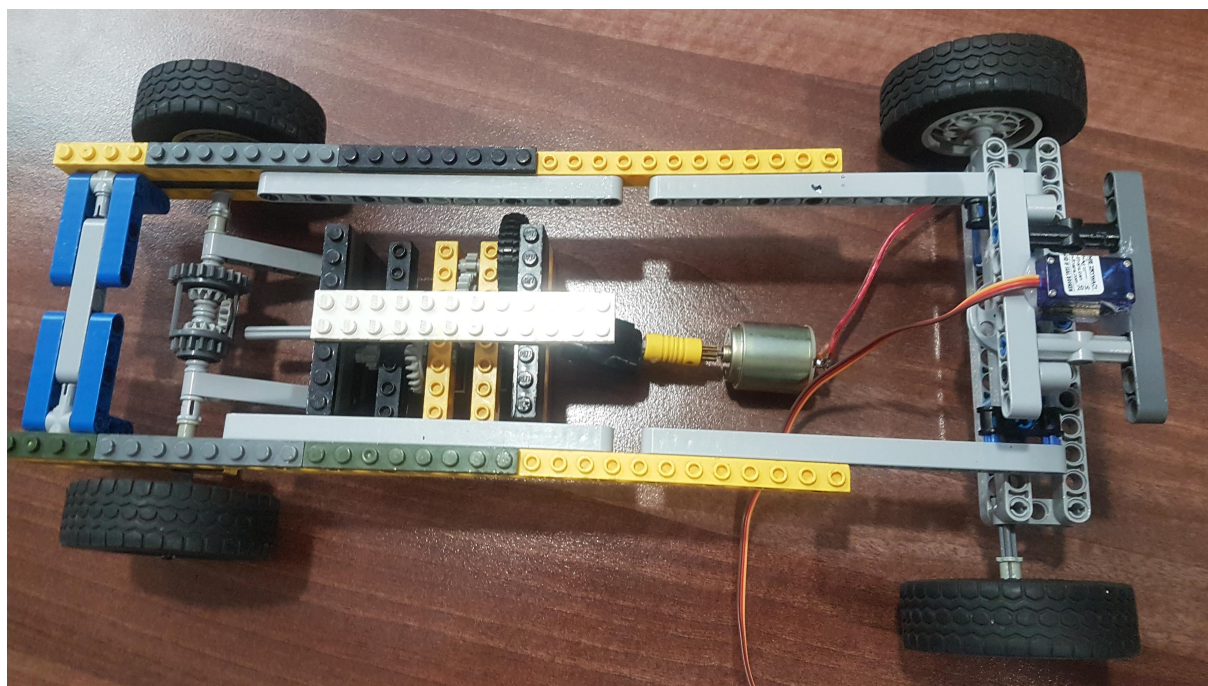
### 3.2 Regulacja prędkości napędu

Za pomocą potencjometru regulujemy wypełnienie sygnału PWM. Sygnał ten jest wzmacniany za pomocą tranzystora NPN i przekazywany do silnika DC.

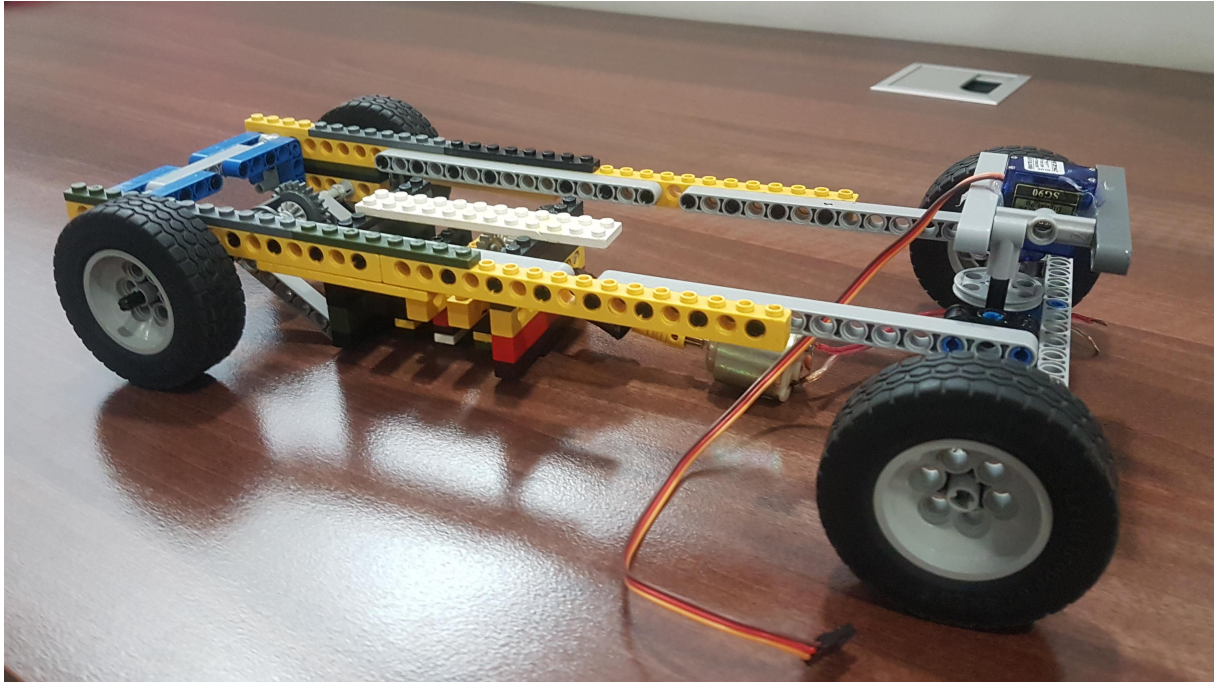


Rysunek 2: Schemat poglądowy regulacji prędkości obrotowej silnika

#### 4 Konstrukcja mechaniczna



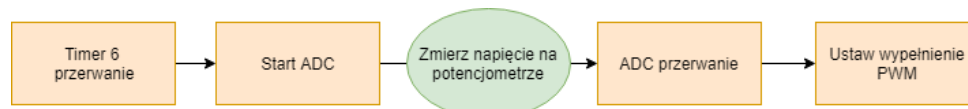
Rysunek 3: Zdjęcie części mechanicznej nr 1



Rysunek 4: Zdjęcie części mechanicznej nr 2

## 5 Opis działania programu

### 5.1 Schemat działania programu



Rysunek 5: 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
           lewo/prawo
    }
  
```

```

6     if (sposition > 90) {
7         //jeśli maksymalny odczyt z lewej strony
8         if ((pos == 0) || (pos == 2))
9         {
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 Zadania niezrealizowane

Nie zostało zrealizowane przekazanie napędu z silnika i serwomechanizmu na mechanizm napędowy oraz na ten służący do skręcania. W pierwszym przypadku jest to spowodowane brakiem czasu, wynikający ze zbyt długim poszukiwaniem rozwiązania na problem przeniesienia napędu z silnika do przekładni, natomiast w drugim tym, że wał serwomechanizmu ma stępione zębatki co uniemożliwia przekazanie jakiegokolwiek siły na dalszy podzespół.

## 7 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